Express Mail Label No. EL601700005US

10-02-00

A

jc922 U.S. PTO 09/676714 09/29/00

UTILITY PATENT APPLICATION TRANSMITTAL
(Large Entity)

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Docket No. POU920000130US
Total Pages in this Submission

TO THE ASSISTANT COMMISSIONER FOR PATENTS
Box Patent Application
Washington, D.C. 20231

Transmitted herewith for filing under 35 U.S.C. 111(a) and 37 C.F.R. 1.53(b) is a new utility patent application for an invention entitled:

METHOD, SYSTEM AND PROGRAM PRODUCTS FOR DETERMINING WHETHER I/O CONSTRAINTS EXIST FOR CONTROLLERS OF A COMPUTING ENVIRONMENT

jc913 U.S. PTO 09/676714 09/29/00

and invented by:

William J. Rooney, Peter B. Yocom, Harry M. Yudenfriend

If a CONTINUATION APPLICATION, *check appropriate box and supply the requisite information:*

☐ **Continuation** ☐ **Divisional** ☐ **Continuation-in-part (CIP)** of prior application No.: _____

Which is a:

☐ **Continuation** ☐ **Divisional** ☐ **Continuation-in-part (CIP)** of prior application No.: _____

Which is a:

☐ **Continuation** ☐ **Divisional** ☐ **Continuation-in-part (CIP)** of prior application No.: _____

Enclosed are:

Application Elements

1. ☒ Filing fee as calculated and transmitted as described below
2. ☒ Specification having 57 pages and including the following:
   - a. ☒ Descriptive Title of the Invention
   - b. ☒ Cross References to Related Applications *(if applicable)*
   - c. ☐ Statement Regarding Federally-sponsored Research/Development *(if applicable)*
   - d. ☐ Reference to Microfiche Appendix *(if applicable)*
   - e. ☒ Background of the Invention
   - f. ☒ Brief Summary of the Invention
   - g. ☒ Brief Description of the Drawings *(if drawings filed)*
   - h. ☒ Detailed Description
   - i. ☒ Claim(s) as Classified Below
   - j. ☒ Abstract of the Disclosure

UTILITY PATENT APPLICATION TRANSMITTAL
(Large Entity)

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Docket No.
POU920000130US
Total Pages in this Submission

Application Elements (Continued)

3. ☒ Drawing(s) *(when necessary as prescribed by 35 USC 113)*

   a. ☒ Formal Number of Sheets **Thirteen (13)**

   b. ☐ Informal Number of Sheets _____

4. ☒ Oath or Declaration

   a. ☐ Newly executed *(original or copy)* ☒ Unexecuted

   b. ☐ Copy from a prior application (37 CFR 1.63(d)) *(for continuation/divisional application only)*

   c. ☒ With Power of Attorney ☐ Without Power of Attorney

   d. ☐ *DELETION OF INVENTOR(S)*
   Signed statement attached deleting inventor(s) named in the prior application, see 37 C.F.R. 1.63(d)(2) and 1.33(b).

5. ☐ Incorporation By Reference *(usable if Box 4b is checked)*
The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied under Box 4b, is considered as being part of the disclosure of the accompanying application and is hereby incorporated by reference therein.

6. ☐ Computer Program in Microfiche *(Appendix)*

7. ☐ Nucleotide and/or Amino Acid Sequence Submission *(if applicable, all must be included)*

   a. ☐ Paper Copy

   b. ☐ Computer Readable Copy *(identical to computer copy)*

   c. ☐ Statement Verifying Identical Paper and Computer Readable Copy

Accompanying Application Parts

8. ☐ Assignment Papers *(cover sheet & document(s))*

9. ☐ 37 CFR 3.73(B) Statement *(when there is an assignee)*

10. ☐ English Translation Document *(if applicable)*

11. ☒ Information Disclosure Statement/PTO-1449 ☒ Copies of IDS Citations

12. ☐ Preliminary Amendment

13. ☒ Acknowledgment postcard

14. ☒ Certificate of Mailing

    ☐ First Class ☒ Express Mail *(Specify Label No.):* **EL601700005US**

UTILITY PATENT APPLICATION TRANSMITTAL
(Large Entity)

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Docket No. POU920000130US
Total Pages in this Submission

Accompanying Application Parts (Continued)

15. ☐ Certified Copy of Priority Document(s) *(if foreign priority is claimed)*

16. ☐ Additional Enclosures *(please identify below):*

Fee Calculation and Transmittal

CLAIMS AS FILED

For	#Filed	#Allowed	#Extra	Rate	Fee
Total Claims	48	- 20 =	28	x $18.00	$504.00
Indep. Claims	4	- 3 =	1	x $78.00	$78.00
Multiple Dependent Claims (check if applicable) ☐					$0.00
				BASIC FEE	$690.00
OTHER FEE *(specify purpose)*					$0.00
				TOTAL FILING FEE	$1,272.00

☐ A check in the amount of to cover the filing fee is enclosed.

☒ The Commissioner is hereby authorized to charge and credit Deposit Account No. **09-0463 (IBM)** as described below. A duplicate copy of this sheet is enclosed.

- ☒ Charge the amount of **$1,272.00** as filing fee.
- ☒ Credit any overpayment.
- ☒ Charge any additional filing fees required under 37 C.F.R. 1.16 and 1.17.
- ☐ Charge the issue fee set in 37 C.F.R. 1.18 at the mailing of the Notice of Allowance, pursuant to 37 C.F.R. 1.311(b).

Blanche E. Schiller
Signature

Dated: **September 29, 2000**

Blanche E. Schiller, Esq.
Reg. No. 35,670
HESLIN & ROTHENBERG, P.C.
5 Columbia Circle
Albany, NY 12203
Telephone: (518) 452-5600
Facsimile: (518) 452-5579

cc:

CERTIFICATE OF MAILING BY "EXPRESS MAIL"

In Re Application of: Rooney et al.

Title: METHOD, SYSTEM AND PROGRAM PRODUCTS FOR DETERMINING WHETHER I/O CONSTRAINTS EXIST FOR CONTROLLERS OF A COMPUTING ENVIRONMENT

Attorney Docket No.: POU920000130US1 (0560.331)

jc913 U.S. PTO
09/671714
09/29/00

"EXPRESS MAIL" MAILING LABEL NO. EL601700005US

Date of Deposit September 29, 2000

I hereby certify that this paper is being deposited with the U.S. Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and addressed to:

BOX PATENT APPLICATION
ASSISTANT COMMISSIONER FOR PATENTS
WASHINGTON, D.C. 20231

Jill K. Becker
(Typed or printed name of person mailing paper or fee)

Jill K Becker
(Signature of person mailing paper or fee)

Enclosures:

* Utility Patent Application Transmittal Letter (3 pages) (in duplicate)
* U.S. Patent Application which includes: Specification (38 pages), 48 Claims (18 pages), Abstract (1 page)
* Thirteen (13) sheets of Formal Drawings
* Declaration and Power of Attorney for Patent Application (unsigned) (4 pages)
* Information Disclosure Citation (1 page) and eight (8) references
* Two (2) Acknowledgment Postcards

09/671714 092900

METHOD, SYSTEM AND PROGRAM PRODUCTS FOR DETERMINING WHETHER I/O CONSTRAINTS EXIST FOR CONTROLLERS OF A COMPUTING ENVIRONMENT

Cross-Reference to Related Applications

This application contains subject matter which is related to the subject matter of the following applications, each of which is assigned to the same assignee as this application. Each of the below listed applications is hereby incorporated herein by reference in its entirety:

"Dynamically Redistributing Shareable Resources Of A Computing Environment To Manage The Workload Of That Environment," Kubala et al., (Docket No. PO9-99-159, 0560.279), Serial No. 09/408,470, filed September 28, 1999;

"Method, System And Program Products For Managing Groups Of Partitions Of A Computing Environment," Kubala et al., (Docket No. PO9-99-146, 0560.276), Serial No. 09/407,391, filed September 28, 1999;

"Method, System And Program Products For Managing Central Processing Unit Resources Of A Computing Environment," Eilert et al., (Docket No. PO9-99-148, 0560.278), Serial No. 09/407,212, filed September 28, 1999;

"Method, System And Program Products For Managing Logical Processors Of A Computing Environment," King et al., (Docket No. PO9-99-147, 0560.277), Serial No. 09/407,594, filed September 28, 1999;

"Processing Channel Subsystem Pending I/O Work Queues Based On Priorities," Maergner et al., (Docket No. PO9-99-145, 0560.275), Serial No. 09/407,459, filed September 28, 1999;

"Method, System And Program Products For Determining I/O Configuration Entropy," William J. Rooney, (Docket No. PO9-99- 157, 0560.281), Serial No. 09/407,453, filed September 28, 1999;

"Method And Apparatus For Assigning Resources To Logical Partition Clusters," Rooney et al., (Docket No. PO9-99-063), Serial No. 09/407,810, filed September 28, 1999;

"Method And Apparatus For Creating And Identifying Logical Partition Clusters," Rooney et al., (Docket No. PO9-99-156), Serial No. 09/407,514, filed September 28, 1999;

"Method, System and Program Products For Managing I/O Configurations Of A Computing Environment," Cwiakala et al., (Docket No. PO9-99-158, 0560.280), filed September 28, 1999;

"Measuring Utilization Of Individual Components Of Channels," Glassen et al., (Docket No. POU9-1999-0176-US1, 0560.287), Serial No. 09/539,024, filed March 30, 2000; and

"Method, System And Program Products For Projecting The Impact Of Configuration Changes On Controllers," Johnson et al., (Docket No. POU920000175US1, 0560.353), Serial No. _____, filed herewith.

Technical Field

This invention relates, in general, to input/output (I/O) processing, and in particular, to determining whether I/O constraints exist for controllers of a computing environment.

Background Art

The management of workload plays an important role in computing environments today. Thus, various aspects of processing within a computing environment are scrutinized to ensure a proper allocation of resources and to determine whether any constraints exist. One type of processing that is scrutinized is I/O processing.

In I/O processing, workload management includes properly allocating channels to control units. To determine whether the allocation is sufficient, channel delay is measured. Previously, the metric of channel delay included measuring the response times.

The use of response times in determining channel delay, however, has not been as accurate, as desired. This is because it is not always known what constitutes a good

response time. The value of a response time and whether or not it is satisfactory depends on various factors, including the type of input/output that is being performed (e.g., whether it is transferring large or small amounts of data).

Thus, a need still exists for a more accurate metric of channel delay. A further need exists for a capability that more accurately determines when a controller constraint exists.

Summary of the Invention

The shortcomings of the prior art are overcome and additional advantages are provided through the provision of a method of determining whether I/O constraints exist for controllers of a computing environment. The method includes, for instance, determining an I/O velocity for a controller of the computing environment; and using the I/O velocity to determine whether an I/O constraint exists for the controller.

System and computer program products corresponding to the above-summarized methods are also described and claimed herein.

Advantageously, one or more aspects of the present invention provide a capability for accurately determining delay associated with a resource, such as channel delay. Further, a capability is provided for using this information

to determine whether a constraint exists within the computing environment.

Additional features and advantages are realized through the techniques of the present invention. Other embodiments and aspects of the invention are described in detail herein and are considered a part of the claimed invention.

Brief Description of the Drawings

The subject matter which is regarded as the invention is particularly pointed out and distinctly claimed in the claims at the conclusion of the specification. The foregoing and other objects, features, and advantages of the invention are apparent from the following detailed description taken in conjunction with the accompanying drawings in which:

FIG. 1a depicts one example of a computing environment incorporating and using one or more aspects of the present invention;

FIG. 1b depicts another embodiment of a computing environment incorporating and using one or more aspects of the present invention;

FIG. 2 depicts additional components of the computing environments of FIGs. 1a and 1b incorporating and using aspects of the present invention;

FIG. 3 depicts one embodiment of the overall logic associated with determining controller I/O velocity and using the controller I/O velocity to determine whether a controller constraint exists, in accordance with an aspect of the present invention;

FIGs. 4a-4b depict one embodiment of the logic associated with the collection of data of FIG. 3 and with determining a default I/O velocity, in accordance with an aspect of the present invention;

FIGs. 5a-5b depict one embodiment of a global data structure located within a coupling facility used to store the collected data, in accordance with an aspect of the present invention;

FIG. 6 depicts one embodiment of the logic associated with the storing of data of FIG. 3 and with calculating an actual I/O velocity, in accordance with an aspect of the present invention;

FIG. 7 depicts one embodiment of the logic associated with projecting an adjusted I/O velocity, in accordance with an aspect of the present invention;

FIG. 8 depicts one embodiment of the logic associated with the project I/O velocity invocation

of FIG. 7, in accordance with an aspect of the present invention;

FIG. 9 depicts one embodiment of the logic associated with the project channel utilization of FIG. 8, in accordance with an aspect of the present invention;

FIG. 10 depicts one embodiment of the logic associated with calculating adjusted control unit loads, in accordance with an aspect of the present invention; and

FIG. 11 depicts one embodiment of the logic associated with spreading the load, in accordance with an aspect of the present invention.

Best Mode for Carrying Out the Invention

In accordance with an aspect of the present invention, a capability is provided for determining whether a controller (e.g., a control unit) of a computing environment needs additional resources, such as input/output (I/O) resources (e.g., one or more channels). In making this determination, a metric representing the resource is calculated, as described herein.

One embodiment of a computing environment incorporating and using one or more aspects of the present invention is described with reference to FIG. 1a. A computing

environment 100 is based, for instance, on the Enterprise Systems Architecture (ESA)/390 offered by International Business Machines Corporation, Armonk, New York. ESA/390 is described in an IBM publication entitled Enterprise Systems Architecture/390 Principles Of Operation, IBM Publication No. SA22-7201-06, July 1999, which is hereby incorporated herein by reference in its entirety. One example of a computing environment based on ESA/390 is the 9672 Parallel Enterprise Server offered by International Business Machines Corporation.

Computing environment 100 includes, for example, a central processor complex (CPC) 102 having one or more central processors 106 (e.g., CP1-CP4), one or more partitions 108 (e.g., logical partitions (LP1-LP4)), and at least one logical partition manager 110, each of which is described below.

Central processors 106 are physical processor resources that are allocated to the logical partitions. In particular, each logical partition 108 has one or more logical processors (not separately shown for clarity), each of which represents all or a share of a physical processor 106 allocated to the partition. The logical processors of a particular partition 108 may be either dedicated to the partition (so that the underlying processor resource 106 is reserved for that partition) or shared with another partition (so that the underlying processor resource is potentially available to another partition).

In the particular example shown, each of logical partitions LP1-LP4 functions as a separate system having a resident operating system 112 (which may differ for each logical partition) and one or more applications 114. In one embodiment, operating system 112 is the OS/390™ operating system offered by International Business Machines Corporation.

Additionally, each operating system (or a subset thereof) includes a workload manager 116 for managing the workload within a partition and among partitions. One example of a workload manager is WLM offered by International Business Machines Corporation. WLM is described in, for instance, U.S. Patent No. 5,473,773, Aman et al., entitled "Apparatus And Method For Managing A Data Processing System Workload According To Two Or More Distinct Processing Goals", issued December 5, 1995; and U.S. Patent No. 5,675,739, Eilert et al., entitled "Apparatus And Method For Managing A Distributed Data Processing System Workload According To A Plurality Of Distinct Processing Goal Types", issued October 7, 1997, each of which is hereby incorporated herein by reference in its entirety.

Logical partitions 108 are managed by logical partition manager 110 implemented by microcode running on processors 106. Logical partitions 108 (LP1-LP4) and logical partition manager 110 each comprise one or more programs residing in respective portions of central storage associated with the central processors. One example of logical partition manager 110 is the Processor Resource/Systems Manager™

(PR/SM™), which is described, for instance, in the IBM publication Processor Resource/Systems Manager Planning Guide, GA22-7236-04, March 1999, which is hereby incorporated herein by reference in its entirety.

Examples of logically partitioned computing systems are described in, for instance, Guyette et al., U.S. Patent No. 4,564,903, entitled "Partitioned Multiprocessor Programming System", issued on January 14, 1986; Bean et al., U.S. Patent No. 4,843,541, entitled "Logical Resource Partitioning Of A Data Processing System", issued on June 27, 1989; and Kubala, U.S. Patent No. 5,564,040, entitled "Method And Apparatus For Providing A Server Function In A Logically Partitioned Hardware Machine", issued on October 08, 1996, each of which is hereby incorporated herein by reference in its entirety.

In a further embodiment of a computing environment, two or more central processor complexes are coupled to one another to form a sysplex, as depicted in FIG. 1b. As one example, a central processor complex (CPC) 102 is coupled to one or more other CPCs 120 via, for instance, a coupling facility 122.

In the example shown, CPC 120 includes a plurality of logical partitions 124 (e.g., LP1-LP3), which are managed by a logical partition manager 126. One or more of the logical partitions include an operating system, which may have a workload manager and one or more application programs (not shown in this example for clarity). Additionally, CPC 120

includes a plurality of central processors 128 (e.g., CP1-CP3), the resources of which are allocated among the plurality of logical partitions. In particular, the resources are allocated among one or more logical processors 130 of each partition. (In other embodiments, each CPC may have one or more logical partitions and one or more central processors.)

Coupling facility 122 (a.k.a., a structured external storage (SES) processor) contains storage accessible by the central processor complexes and performs operations requested by programs in the CPCs. (In one embodiment, each central processor complex is coupled to a plurality of coupling facilities.) Aspects of the operation of a coupling facility are described in detail in such references as Elko et al., U.S. Patent No. 5,317,739 entitled "Method And Apparatus For Coupling Data Processing Systems", issued May 31, 1994; Elko et al., U.S. Patent No. 5,561,809, entitled "In A Multiprocessing System Having A Coupling Facility Communicating Messages Between The Processors And The Coupling Facility In Either A Synchronous Operation Or An Asynchronous Operation", issued October 1, 1996; Elko et al., U.S. Patent No. 5,706,432, entitled "Mechanism For Receiving Messages At A Coupling Facility", issued January 6, 1998; and the patents and applications referred to therein, all of which are hereby incorporated herein by reference in their entirety.

In one embodiment, one or more of the central processors are coupled to at least one channel subsystem,

which is used in communicating with I/O devices. For example, a central processor 200 (FIG. 2) is coupled to main storage 202 and at least one channel subsystem 204. Channel subsystem 204 is further coupled to one or more control units 206. The control units are then coupled to one or more I/O devices 208.

The channel subsystem directs the flow of information between the input/output devices and main storage. It relieves the central processing units of the task of communicating directly with the input/output devices and permits data processing to proceed concurrently with input/output processing. The channel subsystem uses one or more channel paths 214 as communication links in managing the flow of information to or from input/output devices 208.

Each channel path 214 includes, for instance, a channel 210 of channel subsystem 204, a control unit 206 and a link 212 between the channel and control unit. In other embodiments, a channel path may have multiple channels, control units, and/or links. Further, in another example, it is also possible to have one or more dynamic switches as part of the channel path. A dynamic switch is coupled to a channel and a control unit and provides the capability of physically interconnecting any two links that are attached to the switch. Further details regarding channel subsystems are described in Casper et al., U.S. Patent No. 5,526,484, entitled "Method And System For Pipelining The Processing Of Channel Command Words," issued on June 11, 1996; and Glassen et al., Serial No. 09/539,024, entitled "Measuring

Utilization Of Individual Components Of Channels," filed March 30, 2000, each of which is hereby incorporated herein by reference in its entirety.

The number of resources (e.g., channels) that are available for use by a controller (e.g, a control unit) can be dynamically adjusted to address changing requirements of the workload. In order to determine what adjustments are to be made, a metric of resource delay is employed. In one example, this metric is referred to as I/O velocity. The I/O velocity indicates, for instance, channel delay. It can be used, in one example, to determine whether a constraint (e.g., of communications bandwidth, such as channel bandwidth) exists on a controller (e.g., a control unit (CU)).

In accordance with an aspect of the present invention, an actual I/O velocity of a control unit is calculated using the following equation:

Control Unit I/O Velocity = ΣDevice Connect Times/ (ΣDevice Connect Times + Σ(Pending Times - (CU Busy Times + Device Busy Times))),

in which each device connect time reflects the amount of time that a channel of the control unit is actually being used; and the pending time indicates the time the control unit is waiting to use the channel because all of the channels the control unit has access to (e.g., up to 8 channels) are busy. The pending time includes a number of

other busy indicators, including for instance, control unit busy and device busy, which are removed from the pending time, in this embodiment.

The various components of the control unit I/O velocity equation and use of the control unit I/O velocity are described in further detail with reference to FIGs. 3-6. In particular, FIG. 3 depicts an overview of the steps performed to determine whether a constraint on a controller exists, and FIGs. 4-6 provide additional details for those steps.

Referring to FIG. 3, initially, data is collected from the systems that access the control unit, STEP 300 (FIG. 3). For example, if the computing environment is a partitioned environment, then data is collected from all of the partitions that access the control unit. In one instance, the data is collected by an I/O Supervisor (IOS) of the operating system.

The I/O Supervisor then passes the data to a workload manager (e.g., WLM), which stores the data in, for instance, a coupling facility, STEP 302. This storing of the data in the coupling facility allows the creation of a clustered LPAR-wide view of the data used to calculate the actual I/O velocity. (An LPAR cluster is, for instance, a set of partitions running on a single CPC that are all members of the same sysplex.)

The stored data, which includes the actual control unit I/O velocity, is then capable of being used to determine whether a constraint exists for a control unit of the computing environment, STEP 304.

One embodiment of collecting the data is described in further detail with reference to FIGs. 4a-4b. The logic of FIGs. 4a-4b is performed by IOS, when invoked by, for instance, a workload manager at predefined intervals (e.g., every ten seconds). When the logic is invoked, some initialization is performed, STEP 400. For instance, a workarea in local memory of the system running the logic is cleared, and two variables, I/O Velocity Total and Connect Time Total, are set to zero.

Thereafter, since data is collected for each device of each control unit of a cluster, a control unit is selected, STEP 401. In particular, in one instance, a subsystem control block (SSCB) representing a control unit is selected. Thus, the first subsystem control block on a chain of SSCBs is selected, which identifies the control unit for which data is being collected.

Subsequently, a device of that control unit is selected for processing, STEP 402. In one example, the list of devices for a control unit is obtained using its associated SSCB. Then, pertinent data for that device is obtained, STEP 404. For instance, the pending time and connect time are extracted from a measurement block (a.k.a., a channel measurement block (CMB)) updated by its associated

channel(s). The channel measurement block keeps accumulated data of various times, including device connect time and pending time.

Additionally, the control unit busy time and device busy time are obtained via, for instance, issuing a Store Subchannel instruction (STSCH). Examples of a measurement block and a Store Subchannel instruction are described in the aforementioned Enterprise Systems Architecture/390 Principles of Operation, which has been incorporated herein by reference in its entirety.

Running totals of pending time, device connect time, control unit busy time and device busy time are maintained for the devices of the current control unit, STEP 406. Thus, the pending time retrieved for the current device is added to the sum of pending times for any other devices of this control unit already processed. Likewise, the device connect time for this device is added to a total device connect time for the control unit; the control unit busy time is added to a total control unit busy; and a device busy time is added to a total device busy.

Next, a determination is made as to whether there are more devices for this SSCB, INQUIRY 408. If there are more devices, then processing continues with STEP 402, as described above. However, if there are no more devices for this SSCB, then processing proceeds with calculating new deltas, STEP 410. In particular, since each of the retrieved values represents an accumulation of values, a

calculation is performed to determine how much has changed since the last calculation. For example, the last value for a particular component (e.g., device connect time, pending time, control unit busy time and device busy time) stored in the SSCB is subtracted from its corresponding new value to determine the delta.

Thereafter, the data is stored, for instance, in the coupling facility in order to provide in this example an LPAR cluster-wide view of the data, STEP 412. One example of storing the data in the coupling facility is described further below with reference to FIG. 6.

Subsequent to storing the data in the coupling facility, the SSCB is updated with the new last values, STEP 414. In particular, the last values for the device connect time, pending time, control unit busy time and device busy time are stored in the SSCB.

After updating the SSCB with the new last values, a determination is made as to whether this SSCB supports managed channel paths, INQUIRY 416 (FIG. 4b). In one example, this determination is made by checking a mask in the SSCB. If this control unit supports managed channel paths, then a further determination is made as to whether a target I/O velocity is specified in the SSCB, INQUIRY 418. If managed channel paths are supported and no target I/O velocity is specified, then this SSCB is used to calculate a default I/O velocity. In order to make this calculation, first the I/O velocity returned from the service writing the

data into the coupling facility, as described below, is multiplied by an average connect time returned, STEP 420. The result of that multiplication is then added to the I/O velocity total, STEP 422. Additionally, the average connect time is added to the connect time total, STEP 424.

Subsequently, a determination is made as to whether there are more control units to be processed, INQUIRY 426. If there are more control units to be processed, then processing continues with STEP 401 (FIG. 4a) selecting a control unit. However, if there are no more control units to be processed, then the default I/O velocity to be used by all the control units that do not indicate a target I/O velocity (and support managed channels) is calculated, STEP 428. In one example, the default I/O velocity is equal to the I/O velocity total divided by the connect time total. (In the case where either no controllers had managed channels or all have an explicit target I/O velocity, then the default is set to a value, such as 100.) This default is then stored in a control block that is commonly accessible to IOS, STEP 430. In one example, this commonly accessible control block is referred to as a Dynamic CHPID Management Vector Table (DCMVT).

As described above, in order to provide an LPAR cluster-wide view of the data, the data is stored in the coupling facility. In one example, various components of the data are stored in a global data structure located within the coupling facility, as shown in FIGs. 5a-5b. In one example, there is one global data structure per SSCB.

The global data structure contains device connect times for a plurality of intervals (e.g., twelve). This enables decisions to be based on a moving average.

A global data structure 500 includes, for instance, an index 502 used to indicate the current interval; a control unit I/O velocity field 504 for storing the calculated actual control unit I/O velocity; a plurality of connect times 506, one for each of the plurality of intervals (e.g., intervals 1-12); an average of connect time intervals field 508, which indicates an average of eleven of the twelve intervals; a plurality of pending times 510, one for each of the plurality of intervals; an average of pending time intervals field 512 that includes an average of eleven of the twelve pending time intervals; and a plurality of control unit plus device busy times 514, one for each interval; and an average of CU + device busy intervals field 516, which indicates an average of eleven of the twelve CU + device busy intervals.

One embodiment of the logic associated with storing data in the coupling facility is described with reference to FIG. 6. Initially, a time stamp is obtained, STEP 600. The time from the time stamp is then converted to tens of seconds, STEP 602. Thereafter, an index is calculated by taking modulo 12 of the time in tens, STEP 604.

Subsequently, the global data structure of the coupling facility is read into local memory and the index that was just calculated is compared to the index stored in the

global data structure, STEP 606. If the calculated index is equal to the stored index indicating that a new interval is not being started, INQUIRY 608, then the pertinent data is added to the global data structure, STEP 610. In particular, the connect time, the pending time, the control unit busy time and the device busy time are added to the values already stored at those fields in the appropriate interval.

However, if the calculated index is not equal to the stored index, this indicates that a new interval is being started. Thus, certain information is calculated for the current interval that recently ended. For example, an average connect time is calculated using 11 of the 12 intervals (the new interval is left out), STEP 612, and that average is stored at average of intervals field 508. Similarly, an average of pending time is calculated using 11 of the 12 intervals, and that average is stored at average of pending time intervals 512; and an average of CU + busy time is calculated and stored in field 516. Additionally, a new control unit I/O velocity is calculated and stored at field 504, STEP 614. In particular, the actual I/O velocity is determined by using the control unit I/O velocity equation described above. In that equation, ΣDevice Connect Times is set equal to the average of intervals 508, ΣPending Times is set to average of pending time intervals 512, and ΣCU Busy Times and Device Busy Times is set to average of CU and device busy intervals 516.

Further, the new index that was calculated at STEP 604 is now stored at field 502, STEP 616. Additionally, the new data (e.g., the connect time, control unit and device busy times, and pending time) are stored in their respective fields for the new interval, STEP 618.

Thereafter, a version number associated with the data structure is incremented, and a writing of the data structure to the coupling facility from local memory is initiated, STEP 620. During this initiation, if the version number check fails, INQUIRY 622, then processing continues with STEP 606 and the data structure is not written to the coupling facility. Otherwise, the global data structure is written to the coupling facility. Further, the I/O velocity and average connect time are returned to the caller.

At this point, an actual control unit I/O velocity has been determined for each of the desired control units. Each control unit I/O velocity provides a metric of channel delay, which may be used to set channel bandwidth consumption targets for individual control units. In one example, the target for a control unit is the default calculated above or an explicit target specified by WLM. Further, in another embodiment, a fixed value may be used.

A balance checking service may then be invoked by the workload manager at predetermined intervals (e.g., every ten seconds) (or the logic of FIGs. 4a-4b automatically flow into the balance checking service) to determine if any of the control units are not achieving their target I/O

velocity. In particular, if any of the control units are found outside of a tolerance of, for instance, plus or minus 5 percentage points of the target I/O velocity, an imbalance correction function is executed in an attempt to place the control unit within range. One embodiment of an imbalance correction technique is described in detail in Cwiakala et al., Serial No. 09/407,544, entitled "Method, System and Program Products For Managing I/O Configurations Of A Computing Environment," filed September 28, 1999, which is hereby incorporated herein by reference in its entirety.

In the imbalance correction technique, each control unit failing to meet the target, referred to herein as the target control unit, has associated therewith one or more decision selection blocks (DSBs), each of which represents a different possible solution (e.g., a different configuration). Further, each DSB includes a list of one or more control units that would be affected by the change represented by that DSB.

For the target control unit and each affected control unit of a DSB (collectively referred to herein as affected control units), a projected adjusted I/O velocity technique is invoked, in accordance with an aspect of the present invention, in order to determine a projected adjusted I/O velocity for each of the affected control units of the DSB should the configuration be adopted. The projected I/O velocities resulting from the invocation of the routine are stored in the DSB. This information is then used to select

the best option for the target control unit within the tolerance of the target I/O velocity.

One embodiment of the logic associated with projecting adjusted I/O velocities for affected control units of a DSB is described with reference to FIG. 7. Initially, a project I/O velocity routine (described below) is invoked in order to project the I/O velocity for the control units based on the current I/O configuration, STEP 700. That is, the project I/O velocity routine is executed for the current configuration in order to obtain one or more a base numbers (i.e., one or more base I/O velocities) to be used later on for comparison. Subsequent to projecting the I/O velocity for the current I/O configuration, the project I/O velocity routine is invoked for the control units, once again. This time it is executed for a proposed I/O configuration associated with the DSB, STEP 702. Thereafter, a projected adjusted I/O velocity for each control unit of the DSB is determined using the projected I/O velocity values obtained from the two executions of project I/O velocity, STEP 704. In one example, the projected adjusted I/O velocity for a particular control unit is obtained by calculating the delta of the two values (i.e., the project I/O velocity of the proposed configuration - the project I/O velocity of the current configuration) and adding the delta to the actual I/O velocity of the current configuration previously determined for the control unit. This value is placed in the DSB corresponding to the control unit.

One example of the details associated with projecting I/O velocities for affected control units is described with reference to FIG. 8. Initially, the utilization of the channels associated with a configuration (initially for the current configuration and later for the proposed configuration of the target control unit) are projected, STEP 800. To project the channel utilizations for the configuration, the corresponding I/O topology is looked at and the adjusted control unit load on the affected control units, described below, is used.

In one embodiment, the project channel utilization logic processes control units with the least number of channels first. In case of multiple control units with the same number of channels, those control units with the least load (i.e., total connect time) are processed first. However, the logic has no order dependency on processing the channels.

Further details regarding the projecting of channel utilizations is described with reference to FIG. 9. In the description of this logic, a "." notation is used, which indicates a value for a particular component specified after the ".". For instance, failed load.CU indicates there is a failed load value for each control unit of interest.

Referring to FIG. 9, initially, for each control unit affected by the configuration, a variable referred to as old load.CU is initialized to zero, and another variable referred to as failed load.CU is initialized to an adjusted

control unit load, STEP 900. In order to determine an adjusted control unit load, the logic of FIG. 10 is employed.

Referring to FIG. 10, at predefined intervals, one or more counts are taken, which are used during the calculations. In order to obtain a desired count for a particular interval, a timestamp associated with the interval is determined. For example, a channel time delta is set equal to a channel update time, which is retrieved from a facility (e.g., a Channel Path Measurement Facility (CPMF)), minus the last channel update time, which is stored in memory, STEP 1000. This provides a timestamp for the last interval. For this interval, a channel busy time delta is set equal to a channel busy time (retrieved from CPMF) minus the last channel busy time, STEP 1002. (One example of CPMF is described in Glassen et al., Serial No. 09/539,024, entitled "Measuring Utilization Of Individual Components Of Channels", filed March 30, 2000, which is hereby incorporated herein by reference in its entirety.)

Additionally, a channel connect update time delta is set equal to the channel connect update time minus the last channel connect update time to obtain an interval for channel connect time. Then, a channel connect time delta is set equal to a channel connect time minus the last channel connect time, which provides an amount of connect time associated with the data transfer that occurred in the last interval. (In one example, the connect time is being accumulated in a table (e.g., a Channel Path Connect Table,

where it is accumulated I/O by I/O from the CMB), and the timestamp is associated therewith.)

Thereafter, a channel ratio is determined by dividing the channel busy time delta by the channel connect time delta, STEP 1008. Using the above calculations, a channel ratio is determined for each channel of each control unit of the current DSB.

Subsequently, for each control unit, a channel utilization ratio is determined, STEP 1010. In one example, the channel utilization ratio is determined by adding the ratios for the channels connected to the control unit to obtain a sum, and dividing the sum by the number of channels connected to the control unit. Additionally, for each control unit, an adjusted control unit load is determined, STEP 1012. In one example, the adjusted control unit load is equal to the average connect time associated with the control unit (i.e., a sum of the 11 intervals of connect times for the control unit divided by 11) multiplied by the channel utilization ratio for the control unit. This provides the adjusted control unit loads employed by the project channel utilization routine.

Returning to FIG. 9, subsequent to initializing the failed load.CU to the adjusted control unit load for all the affected control units of this configuration, processing continues with initializing a variable referred to as path load.CHPID to zero for all paths (e.g., CHPIDs), STEP 902. Moreover, variables referred to as trial factors.CU and

trial factors.CHPID are initialized to zero for all of the control units and channels, respectively, of all of the affected control units, STEP 904.

Next, a control unit is selected to be processed, and for that control unit, a variable referred to as trial load.CU is set equal to the failed (or unassigned) load.CU/number of channel paths for that control unit, STEP 906. Further, the trial factor for the control unit is determined, STEP 908. In one example, trial factors.CU is set equal to (trial load.CU x 100)/(100-trial load.CU). Thereafter, trial factors.CU is added to the total trial factor for each of the subsystem's paths (i.e., trial factors.CHPID), STEP 910.

Subsequently, a determination is made as to whether there are any more control units to process, INQUIRY 912. If there are more control units to process, then processing continues with STEP 906. However, if there are no more control units to process, then a spread load function is invoked, STEP 914. One embodiment of the logic associated with the spread load function is described below with reference to FIG. 11.

The spread load logic is performed for each control unit of the DSB and for each path within the control unit. Thus, initially the control unit is selected, STEP 1100, and a path within the control unit is selected, STEP 1102.

For the selected path within the control unit, the change in utilization for this CHPID is calculated, STEP 1104. One example of this calculation is as follows: Delta Load= ((100-path load.CHPID) x trial factors.CU)/(100+trial factors.CHPID), in which the values of the variables are obtained from the project channel utilization logic.

Thereafter, the new load for the path is calculated by adding the delta load to the new load.CHPID (i.e., new load.CHPID=new load.CHPID + delta load), STEP 1106. Further, the failed load and old load for the control unit are calculated, respectively. For instance, failed load.CU=failed load.CU-delta load, STEP 1108, and old load.CU.CHPID=old load.CU.CHPID + delta load, STEP 1110.

Subsequently, a determination is made as to whether this is the last CHPID attached to this control unit, INQUIRY 1112. If this is not the last CHPID, then processing continues with STEP 1102. However, if this is the last CHPID of the selected control unit, then a determination is made as to whether this is the last control unit in this DSB, INQUIRY 1114. If this is not the last control unit, then processing continues with STEP 1100. However, if this is the last path of the last control unit, then for each path (i.e., CHPID), its respective path load.CHPID is set equal to the new load.CHPID, STEP 1116.

Referring back to FIG. 9, after performing the spread load, a determination is made as to whether the failed load for each control unit of the DSB is minimal (e.g., within

.5%, or approaching zero), INQUIRY 916. If the failed load for each control unit is not minimal, then processing continues with STEP 904. Otherwise, processing of the project channel utilization is complete, STEP 918. The path load.CHPID represents the projected channel utilization for the corresponding path.

After completing the project channel utilizations, processing continues with determining contention factors, STEP 802 (FIG. 8). In one example, a contention factor is defined as: Given a control unit with N channels at a given utilization, the contention factor for the control unit is the single channel equivalent utilization yielding an equal probability that an I/O request to that control unit would wait. The contention factors are determined using the projected channel utilizations and the configuration (either the current or proposed configuration).

A contention factor is determined for each control unit in the DSB, one at a time. For example, assume that a Control Unit 220 is connected to 4 CHPIDs having the following projected utilizations:

CHPID	UTILIZATION
20	40%
37	42%
48	29%
73	33%

In one example, the CHPIDs connected to the control unit are sorted from lowest to highest projected utilization.

CHPID	UTILIZATION
48	29%
73	33%
20	40%
37	42%

Then, average utilizations are calculated based on the utilization of the first CHPID in the list, then the average of the first two CHPIDs in the list, and so on down the list.

CHPID	UTILIZATION	AVERAGE UTILIZATION
48	29%	29%
73	33%	31%
20	40%	34%
37	42%	36%

Each of the average utilizations is then applied to a table to determine its corresponding contention factor. If the exact utilization is not in the table, then an interpolation is performed. One example of the table is depicted below:

Single CHPID Equivalent Utilization
Yielding Equal Probability Of Waiting
(Contention Factor)

		Number of CHPIDs							
5	**Avg util**	1	2	3	4	5	6	7	8
	0	0	0	0	0	0	0	0	0
	5	5	0	0	0	0	0	0	0
	10	10	2	0	0	0	0	0	0
	15	15	4	1	0	0	0	0	0
10	20	20	7	2	1	0	0	0	0
	25	25	10	4	2	1	0	0	0
	30	30	14	7	4	2	1	1	0
	35	35	18	10	6	4	2	1	1
	40	40	23	14	9	6	4	3	2
15	45	45	28	19	13	9	7	5	3
	50	50	33	24	17	13	10	8	6
	55	55	39	29	23	18	14	12	9
	60	60	45	35	29	24	20	17	14
	65	65	51	42	35	30	26	23	20
20	70	70	58	49	43	38	34	30	27
	75	75	64	57	51	46	42	39	36
	80	80	71	65	60	55	52	49	46
	85	85	78	73	69	65	62	60	57
	90	90	85	82	79	76	74	72	70
25	95	95	93	91	89	88	87	85	84
	100	100	100	100	100	100	100	100	100

The result of the table look-up is:

	CHPID	UTILIZATION	AVERAGE UTILIZATION	CONTENTION FACTOR
30	48	29%	29%	29.0%
	73	33%	31%	14.8
	20	40%	34%	9.4
	37	42%	36%	6.6

The above is repeated for all of the affected control units, yielding contention factors for each control unit of the DSB.

Referring to FIG. 8, after determining the contention factors for the affected control units of the DSB, the contention factors are converted into projected I/O velocities using, for instance, a table per control unit built and managed by a workload manager. The table begins with at least two points: one having an I/O velocity of zero and a contention factor of 100, and another having an I/O velocity of 100 and a contention factor of zero; and it is a running record of control units, and associated actual contention factors and I/O velocities the workload manager has seen in the past. To keep this running record, WLM measures the current I/O velocity and contention factor of each control unit. It then stores the measured I/O velocity in the table entry which represents the contention factor measured. In one embodiment, the table has 50 entries. Each entry in the table represents a contention factor range of 2. The first entry represents contention factors of 0 to 1, the second entry 2 to 3, and so on. If the entry for the current contention factor is empty, WLM stores the measured I/O velocity in the entry. If the entry is not empty, WLM stores a value which is equal to (3* current value + the new I/O velocity)/4. After saving the new I/O velocity, WLM verifies that the rest of the table is consistent with the new I/O velocity. All I/O velocities for higher contention factors should be lower, and I/O velocities for low

contention factors should be higher. WLM clears any table entries that are inconsistent with the new I/O velocity.

Thus, a contention factor is looked up in the table to see historically the result (projected I/O velocity). If the exact value is not in the table, then an interpolation is performed.

So, for the example above, assume the following is determined:

CHPID	UTILIZATION	AVERAGE UTILIZATION	CONTENTION FACTOR	I/O VELOCITY
48	29%	29%	29.0%	(72)
73	33%	31%	14.8	(78)
20	40%	34%	9.4	(83)
37	42%	36%	6.6	95

In one embodiment, the I/O velocity associated with the lowest contention factor is selected to represent the control unit. Thus, in the above example, the lowest contention factor for Control Unit 220, e.g., 6.6, is used as a contention factor for the control unit. Therefore, the I/O velocity for this control unit is 95. The above is repeated for all of the control units of the DSB, and then these projected I/O velocities are returned to project adjusted I/O velocity in FIG. 7 to determine the projected adjusted I/O velocities for the control units of the DSB.

In one example, only the I/O velocity of the lowest contention factor of a control unit needs to be calculated. The others are shown for example purposes.

Described in detail above is a capability for projecting I/O velocities for affected controllers of I/O configuration changes. The projected I/O velocities can be used to determine the best option to be employed to move a controller within its target range.

As described above, one part of the technique is to assign channel path loads. One example of this processing is as follows:

Assume two control units, 4800 and 5600.
Both share the same two CHPIDs, 21 and 22.
CU 4800 has a load of 80, while CU 5600 has a load of 50.
There is an expectation that the CHPID utilizations as equal at (80 + 50)/2 or 65 each.

```
Failed Load.CU4800= 0.800
Failed Load.CU5600= 0.500
Trial Load.CU4800= 0.400
Trial Factors.CU4800= 0.666
Trial Factors.CHPID21= 0.666
Trial Factors.CHPID22= 0.666
Trial Load.CU5600= 0.250
Trial Factors.CU5600=0.333
Trial Factors.CHPID21= 1.000
Trial Factors.CHPID22= 1.000
Delta Load = 0.333
New Load.CHPID21= 0.333
Failed Load.CU4800= 0.466
Old Load.CU4800.CHPID21= 0.333
Delta Load = 0.333
New Load.CHPID22= 0.333
Failed Load.CU4800= 0.133
Old Load.CU4800.CHPID22= 0.333
Delta Load = 0.166
New Load.CHPID21= 0.500
Failed Load.CU5600= 0.333
Old Load.CU5600.CHPID21= 0.166
Delta Load = 0.166
```

```
        New Load.CHPID22=0.500
        Failed Load.CU5600= 0.166
        Old Load.CU5600.CHPID22= 0.166
        Path Load.CHPID21= 0.500
 5      Path Load.CHPID22= 0.500
        --- end of first iteration of major loop
        Trial Load.CU4800 = 0.066
        Trial Factors.CU4800 = 0.071
        Trial Factors.CHPID21 = 0.071
10      Trial Factors.CHPID22 = 0.071
        Trial Load.CU5600 = 0.083
        Trial Factors.CU5600 = 0.090
        Trial Factors.CHPID21 = 0.162
        Trial Factors.CHPID22 = 0.162
15      Delta Load = 0.030
        New Load.CHPID21= 0.530
        Failed Load.CU4800= 0.102
        Old Load.CU4800.CHPID21= 0.364
        Delta Load = 0.030
20      New Load.CHPID22=0.530
        Failed Load.CU4800= 0.071
        Old Load.CU4800.CHPID22=0.364
        Delta Load = 0.039
        New Load.CHPID21= 0.569
25      Failed Load.CU5600= 0.127
        Old Load.CU5600.CHPID21= 0.205
        Delta Load = 0.039
        New Load.CHPID22= 0.569
        Failed Load.CU5600= 0.088
30      Old Load.CU5600.CHPID22= 0.205
        Path Load.CHPID21= 0.569
        Path Load.CHPID22= 0.569
        --- end of second iteration of major loop


        ... and eventually it converges and their result is:
35
```

```
Input Data:
CU4800  800 21 22
CU5600  500 21 22
EOF
          | Paths
          |  21    22
----------+-------------------------------------
CTL Unit  |                Loads
CU4800    |
    80.0  | 40.0  40.0
CU5600    |
    50.0  | 25.0  25.0
----------+-------------------------------------
Totals:   | 65.0  65.0
```

It took 16 iterations to assign the channel-path load. Therefore, the results equal the expectations.

Described in detail above is a capability for projecting the impact of a configuration change on one or more controllers. Although in the embodiment above, the project I/O velocity routine is invoked twice for a DSB, once for the current configuration and once for a proposed configuration, this is only one example. In another example, the routine is only invoked for the proposed configuration. In that example, there may also be no need for determining the projected adjusted I/O velocity. The projected I/O velocity may suffice.

In the embodiments described above, various computing environments and systems are described. These are only examples and are not intended to limit the various aspects of the present invention. Further, various aspects of the present invention are described with reference to logical partitions. The use of logical partitions is only one

example. Aspects of the invention will apply to other types of partitioning, as well as to non-partitioned systems. Thus, these are also considered within the scope of the present invention.

In one example, for non-partitioned systems, storage within a global repository (such as within a coupling facility) is not necessary.

Although an embodiment of the invention is described in terms of a control unit and channels, the invention is not limited to such components. Aspects of the invention are equally applicable to other controllers and resources in other environments, such as storage area networks and other networks. For instance, in further embodiments, the controller is a processor which is communicating with (or transferring data with) another processor (e.g., using channel-to-channel communications); or the controller is a single device (e.g., an I/O device), which has no intelligent control unit between it and a processor. Further, aspects of the invention are equally applicable to environments in which the logic (e.g., for determining whether a constraint exists for a controller) is executed by a controller communicating directly with (or transferring data directly with) the other controller.

The present invention can be included in an article of manufacture (e.g., one or more computer program products) having, for instance, computer usable media. The media has embodied therein, for instance, computer readable program

code means for providing and facilitating the capabilities of the present invention. The article of manufacture can be included as a part of a computer system or sold separately.

Additionally, at least one program storage device readable by a machine, tangibly embodying at least one program of instructions executable by the machine to perform the capabilities of the present invention can be provided.

The flow diagrams depicted herein are just examples. There may be many variations to these diagrams or the steps (or operations) described therein without departing from the spirit of the invention. For instance, the steps may be performed in a differing order, or steps may be added, deleted or modified. All of these variations are considered a part of the claimed invention.

Although preferred embodiments have been depicted and described in detail herein, it will be apparent to those skilled in the relevant art that various modifications, additions, substitutions and the like can be made without departing from the spirit of the invention and these are therefore considered to be within the scope of the invention as defined in the following claims.

Claims

What is claimed is:

1. A method of determining whether I/O constraints exist for controllers of a computing environment, said method comprising:

determining an I/O velocity for a controller of said computing environment; and

using said I/O velocity to determine whether an I/O constraint exists for said controller.

2. The method of claim 1, wherein said determining comprises using the following equation:

$$\text{I/O Velocity} = \Sigma\text{Device Connect Times} / (\Sigma\text{Device Connect Times} + \Sigma(\text{Pending Times} - (\text{Control Unit Busy Times} + \text{Device Busy Times}))).$$

3. The method of claim 2, wherein said ΣDevice Connect Times comprises an average of connect times associated with one or more devices of said controller collected over a predefined amount of time.

4. The method of claim 3, further comprising obtaining said average, said obtaining comprising:

adding connect times of said one or more devices collected over a predefined interval to obtain a sum for the predefined interval;

performing said adding for a plurality of intervals to obtain a plurality of sums; and

summing at least a portion of said plurality of sums to obtain a result and dividing said result by a number representing said at least a portion of said plurality of sums to obtain said average.

5. The method of claim 2, wherein said ΣDevice Pending Times comprises an average of pending times associated with one or more devices of said controller collected over a predefined amount of time.

6. The method of claim 5, further comprising obtaining said average, said obtaining comprising:

adding pending times of said one or more devices collected over a predefined interval to obtain a sum for the predefined interval;

performing said adding for a plurality of intervals to obtain a plurality of sums; and

summing at least a portion of said plurality of sums to obtain a result and dividing said result by a number representing said at least a portion of said plurality of sums to obtain said average.

7. The method of claim 2, wherein said ΣDevice Control Unit Busy + Device Busy Times comprises an average of control unit busy and device busy times associated with one or more devices of said controller collected over a predefined amount of time.

8. The method of claim 7, further comprising obtaining said average, said obtaining comprising:

adding control unit busy and device busy times of said one or more devices collected over a predefined interval to obtain a sum for the predefined interval;

performing said adding for a plurality of intervals to obtain a plurality of sums; and

summing at least a portion of said plurality of sums to obtain a result and dividing said result by a number representing said at least a portion of said plurality of sums to obtain said average.

9. The method of claim 2, wherein said controller is accessible by one or more partitions of a central processing complex, and wherein the device connect times, the pending times and the control unit busy and device busy times reflect data from the one or more partitions.

10. The method of claim 1, wherein said using comprises comparing said I/O velocity to a target I/O velocity to determine whether said I/O constraint exists, wherein said I/O constraint exists when said I/O velocity is not within a tolerance of said target I/O velocity.

11. The method of claim 1, wherein said controller comprises a control unit and said I/O constraint comprises a lack of communications bandwidth.

12. The method of claim 11, wherein the communications bandwidth comprises channel bandwidth.

13. The method of claim 1, wherein said determining comprises employing a relationship between an amount of time waiting to use one or more resources of the controller and an amount of time using the one or more resources to determine the I/O velocity.

14. The method of claim 13, wherein said controller comprises a control unit and said one or more resources comprise one or more channels of the control unit.

15. The method of claim 1, wherein said controller comprises one of the following:

a processor communicating with another processor; and

a single device.

16. A system of determining whether I/O constraints exist for controllers of a computing environment, said system comprising:

means for determining an I/O velocity for a controller of said computing environment; and

means for using said I/O velocity to determine whether an I/O constraint exists for said controller.

17. The system of claim 16, wherein said means for determining comprises means for using the following equation:

I/O Velocity = ΣDevice Connect Times/
(ΣDevice Connect Times + Σ(Pending Times -
(Control Unit Busy Times + Device Busy Times))).

18. The system of claim 17, wherein said ΣDevice Connect Times comprises an average of connect times associated with one or more devices of said controller collected over a predefined amount of time.

19. The system of claim 18, further comprising means for obtaining said average, said means for obtaining comprising:

means for adding connect times of said one or more devices collected over a predefined interval to obtain a sum for the predefined interval;

means for performing said adding for a plurality of intervals to obtain a plurality of sums; and

means for summing at least a portion of said plurality of sums to obtain a result and dividing said result by a number representing said at least a portion of said plurality of sums to obtain said average.

20. The system of claim 17, wherein said ΣDevice Pending Times comprises an average of pending times associated with one or more devices of said controller collected over a predefined amount of time.

21. The system of claim 20, further comprising means for obtaining said average, said means for obtaining comprising:

means for adding pending times of said one or more devices collected over a predefined interval to obtain a sum for the predefined interval;

means for performing said adding for a plurality of intervals to obtain a plurality of sums; and

means for summing at least a portion of said plurality of sums to obtain a result and dividing said result by a number representing said at least a portion of said plurality of sums to obtain said average.

22. The system of claim 17, wherein said ΣDevice Control Unit Busy + Device Busy Times comprises an average of control unit busy and device busy times associated with one or more devices of said controller collected over a predefined amount of time.

23. The system of claim 22, further comprising means for obtaining said average, said means for obtaining comprising:

means for adding control unit busy and device busy times of said one or more devices collected over a predefined interval to obtain a sum for the predefined interval;

means for performing said adding for a plurality of intervals to obtain a plurality of sums; and

means for summing at least a portion of said plurality of sums to obtain a result and dividing said result by a number representing said at least a portion of said plurality of sums to obtain said average.

24. The system of claim 17, wherein said controller is accessible by one or more partitions of a central processing complex, and wherein the device connect times, the pending times and the control unit busy and device busy times reflect data from the one or more partitions.

25. The system of claim 16, wherein said means for using comprises means for comparing said I/O velocity to a target I/O velocity to determine whether said I/O constraint exists, wherein said I/O constraint exists when said I/O velocity is not within a tolerance of said target I/O velocity.

26. The system of claim 16, wherein said controller comprises a control unit and said I/O constraint comprises a lack of communications bandwidth.

27. The system of claim 26, wherein said communications bandwidth comprises channel bandwidth.

28. The system of claim 16, wherein said means for determining comprises means for employing a relationship between an amount of time waiting to use one or more resources of the controller and an amount of time using the one or more resources to determine the I/O velocity.

29. The system of claim 28, wherein said controller comprises a control unit and said one or more resources comprise one or more channels of the control unit.

30. The system of claim 16, wherein said controller comprises one of the following:

a processor communicating with another processor; and

a single device.

09/975,109

31. A system of determining whether I/O constraints exist for controllers of a computing environment, said system comprising:

a first processor to determine an I/O velocity for a controller of said computing environment; and

a second processor to use said I/O velocity to determine whether an I/O constraint exists for said controller.

32. The system of claim 31, wherein the first processor and the second processor are the same processor.

33. The system of claim 31, wherein the first processor and the second processor are different processors.

34. At least one program storage device readable by a machine, tangibly embodying at least one program of instructions executable by the machine to perform a method of determining whether I/O constraints exist for controllers of a computing environment, said method comprising:

determining an I/O velocity for a controller of said computing environment; and

using said I/O velocity to determine whether an I/O constraint exists for said controller.

35. The at least one program storage device of claim 34, wherein said determining comprises using the following equation:

$$\text{I/O Velocity} = \Sigma\text{Device Connect Times} / (\Sigma\text{Device Connect Times} + \Sigma(\text{Pending Times} - (\text{Control Unit Busy Times} + \text{Device Busy Times}))).$$

36. The at least one program storage device of claim 35, wherein said ΣDevice Connect Times comprises an average of connect times associated with one or more devices of said controller collected over a predefined amount of time.

37. The at least one program storage device of claim 36, wherein said method further comprises obtaining said average, said obtaining comprising:

adding connect times of said one or more devices collected over a predefined interval to obtain a sum for the predefined interval;

performing said adding for a plurality of intervals to obtain a plurality of sums; and

summing at least a portion of said plurality of sums to obtain a result and dividing said result by a number representing said at least a portion of said plurality of sums to obtain said average.

38. The at least one program storage device of claim 35, wherein said ΣDevice Pending Times comprises an average of pending times associated with one or more devices of said controller collected over a predefined amount of time.

39. The at least one program storage device of claim 38, wherein said method further comprises obtaining said average, said obtaining comprising:

adding pending times of said one or more devices collected over a predefined interval to obtain a sum for the predefined interval;

performing said adding for a plurality of intervals to obtain a plurality of sums; and

summing at least a portion of said plurality of sums to obtain a result and dividing said result by a number representing said at least a portion of said plurality of sums to obtain said average.

40. The at least one program storage device of claim 35, wherein said ΣDevice Control Unit Busy + Device Busy Times comprises an average of control unit busy and device busy times associated with one or more devices of said controller collected over a predefined amount of time.

41. The at least one program storage device of claim 40, wherein said method further comprises obtaining said average, said obtaining comprising:

adding control unit busy and device busy times of said one or more devices collected over a predefined interval to obtain a sum for the predefined interval;

performing said adding for a plurality of intervals to obtain a plurality of sums; and

summing at least a portion of said plurality of sums to obtain a result and dividing said result by a number representing said at least a portion of said plurality of sums to obtain said average.

42. The at least one program storage device of claim 35, wherein said controller is accessible by one or more partitions of a central processing complex, and wherein the device connect times, the pending times and the control unit busy and device busy times reflect data from the one or more partitions.

43. The at least one program storage device of claim 34, wherein said using comprises comparing said I/O velocity to a target I/O velocity to determine whether said I/O constraint exists, wherein said I/O constraint exists when said I/O velocity is not within a tolerance of said target I/O velocity.

44. The at least one program storage device of claim 34, wherein said controller comprises a control unit and said I/O constraint comprises a lack of communications bandwidth.

45. The at least one program storage device of claim 44, wherein said communications bandwidth comprises channel bandwidth.

46. The at least one program storage device of claim 34, wherein said determining comprises employing a relationship between an amount of time waiting to use one or more resources of the controller and an amount of time using the one or more resources to determine the controller I/O velocity.

47. The at least one program storage device of claim 46, wherein said controller comprises a control unit and said one or more resources comprise one or more channels of the control unit.

48. The at least one program storage device of claim 34, wherein said controller comprises one of the following:

a processor communicating with another processor; and

a single device.

* * * * *

09/09/04, 09:50:00

METHOD, SYSTEM AND PROGRAM PRODUCTS FOR DETERMINING WHETHER I/O CONSTRAINTS EXIST FOR CONTROLLERS OF A COMPUTING ENVIRONMENT

Abstract of the Disclosure

A determination is made as to whether input/output constraints exist for controllers (e.g., control units) of a computing environment. To facilitate this determination, an I/O velocity is calculated. The I/O velocity represents a relationship between an amount of time waiting to use one or more resources of the controller and an amount of time using the one or more resources.

0960744.09090

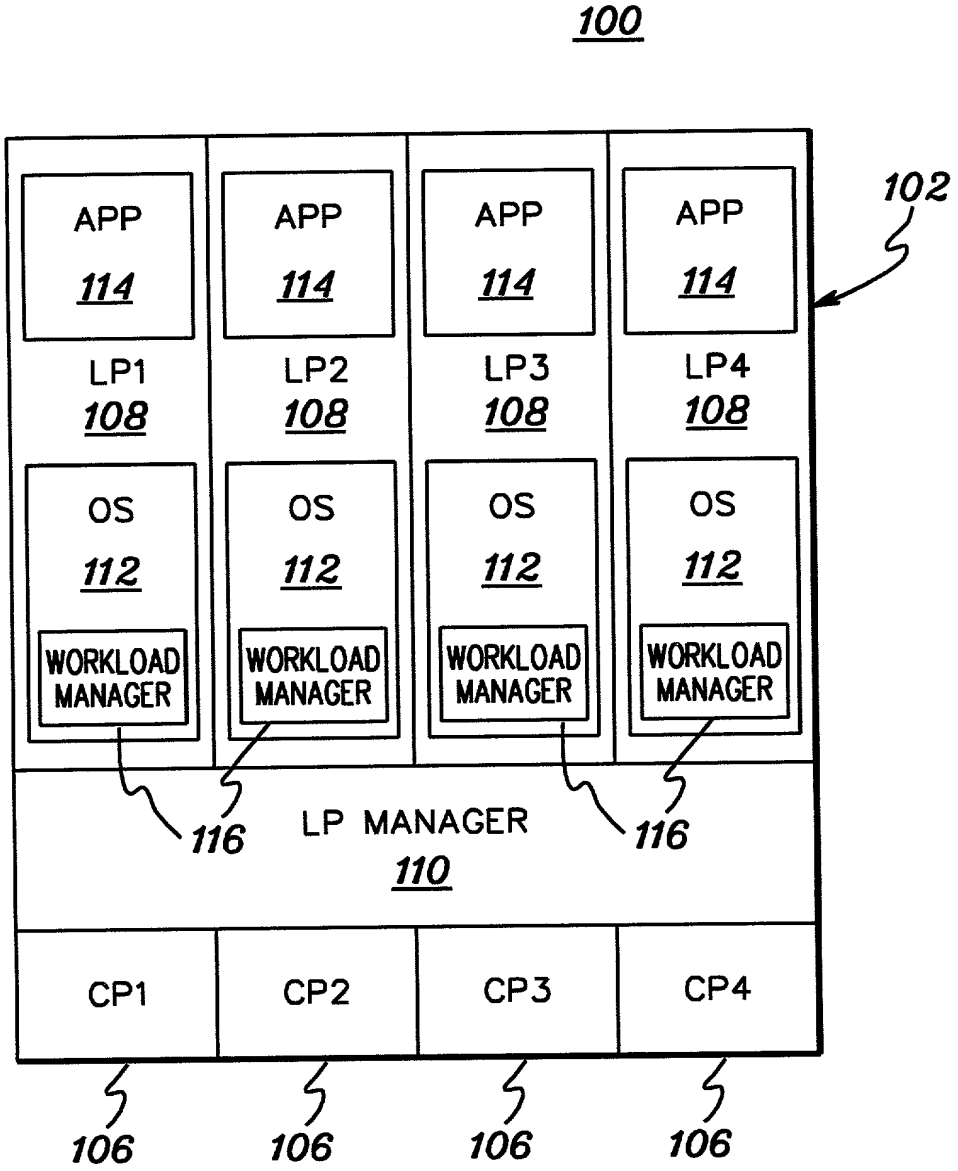
100
102
APP
114
APP
114
APP
114
APP
114
LP1
108
LP2
108
LP3
108
LP4
108
OS
112
OS
112
OS
112
OS
112
WORKLOAD MANAGER
WORKLOAD MANAGER
WORKLOAD MANAGER
WORKLOAD MANAGER
116
LP MANAGER
110
116
CP1
CP2
CP3
CP4
106
106
106
106

fig. 1A

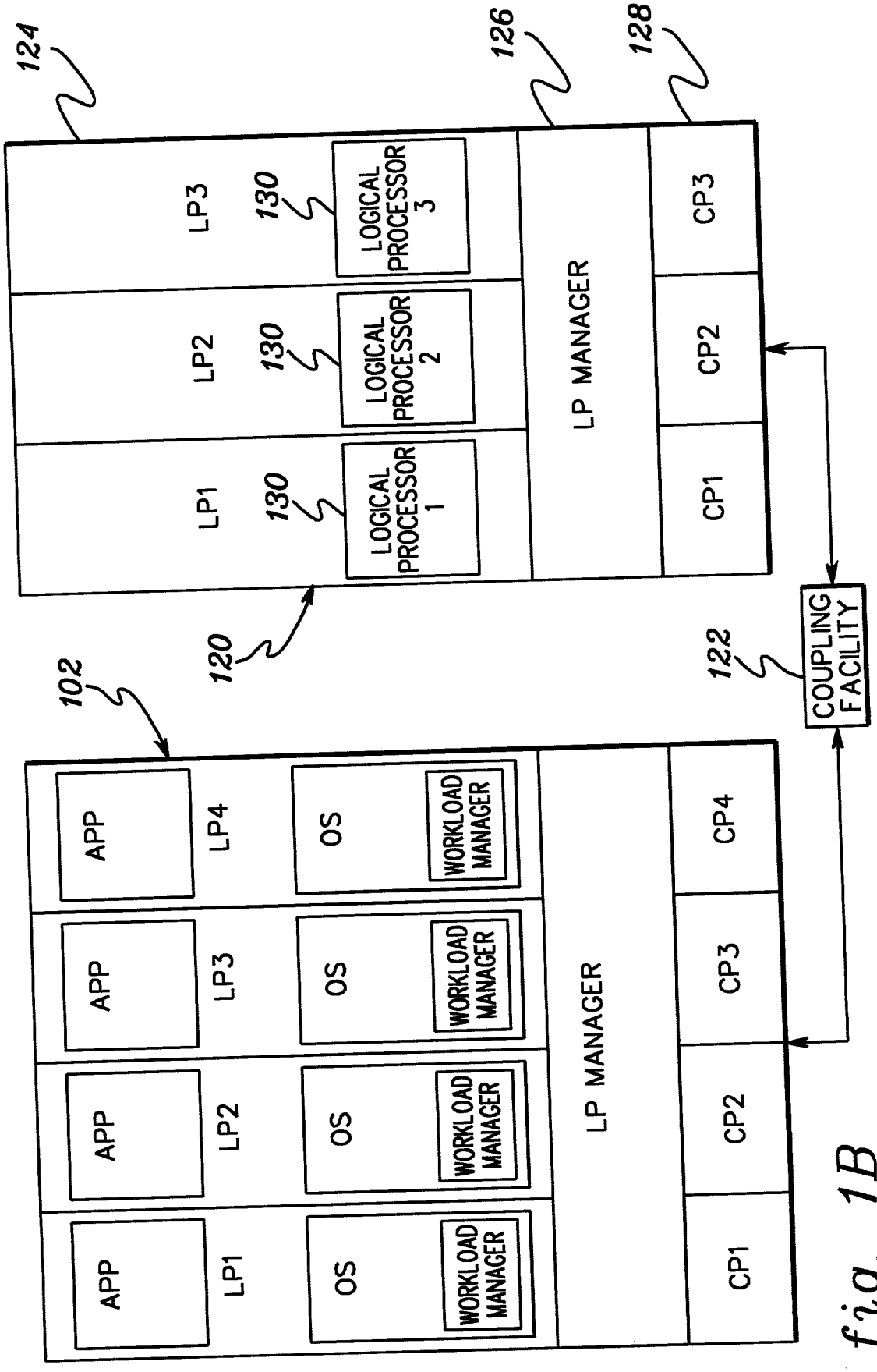
124
126
128
LP3
LP2
LP1
130
LOGICAL PROCESSOR 3
LOGICAL PROCESSOR 2
LOGICAL PROCESSOR 1
LP MANAGER
CP3
CP2
CP1
120
122
COUPLING FACILITY
102
APP
LP4
LP3
LP2
LP1
OS
WORKLOAD MANAGER
LP MANAGER
CP4
CP3
CP2
CP1

fig. 1B

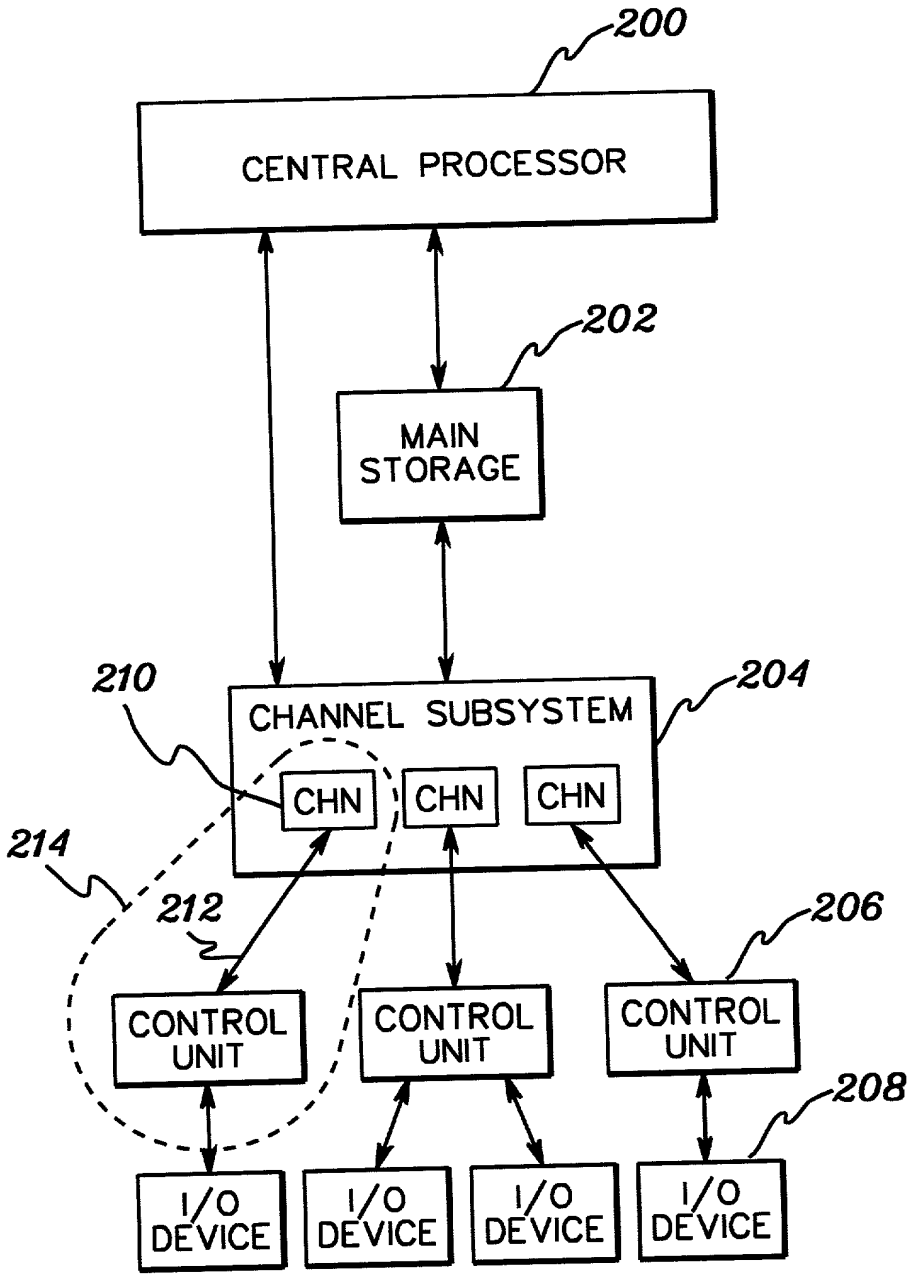
200
CENTRAL PROCESSOR
202
MAIN STORAGE
210
CHANNEL SUBSYSTEM
204
CHN
CHN
CHN
214
212
206
CONTROL UNIT
CONTROL UNIT
CONTROL UNIT
208
I/O DEVICE
I/O DEVICE
I/O DEVICE
I/O DEVICE

fig. 2

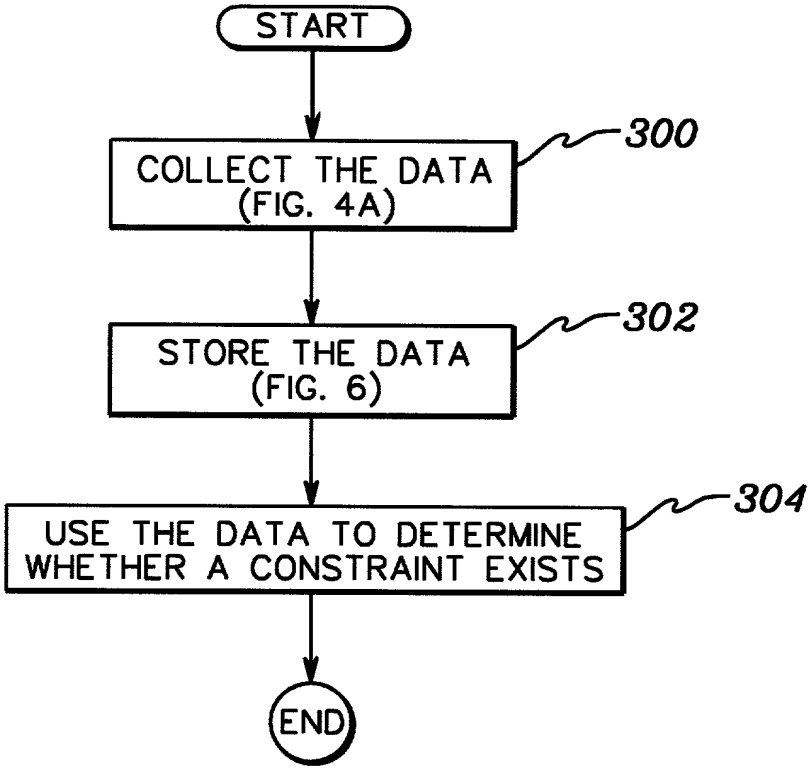
START
COLLECT THE DATA (FIG. 4A)
300
STORE THE DATA (FIG. 6)
302
USE THE DATA TO DETERMINE WHETHER A CONSTRAINT EXISTS
304
END

fig. 3

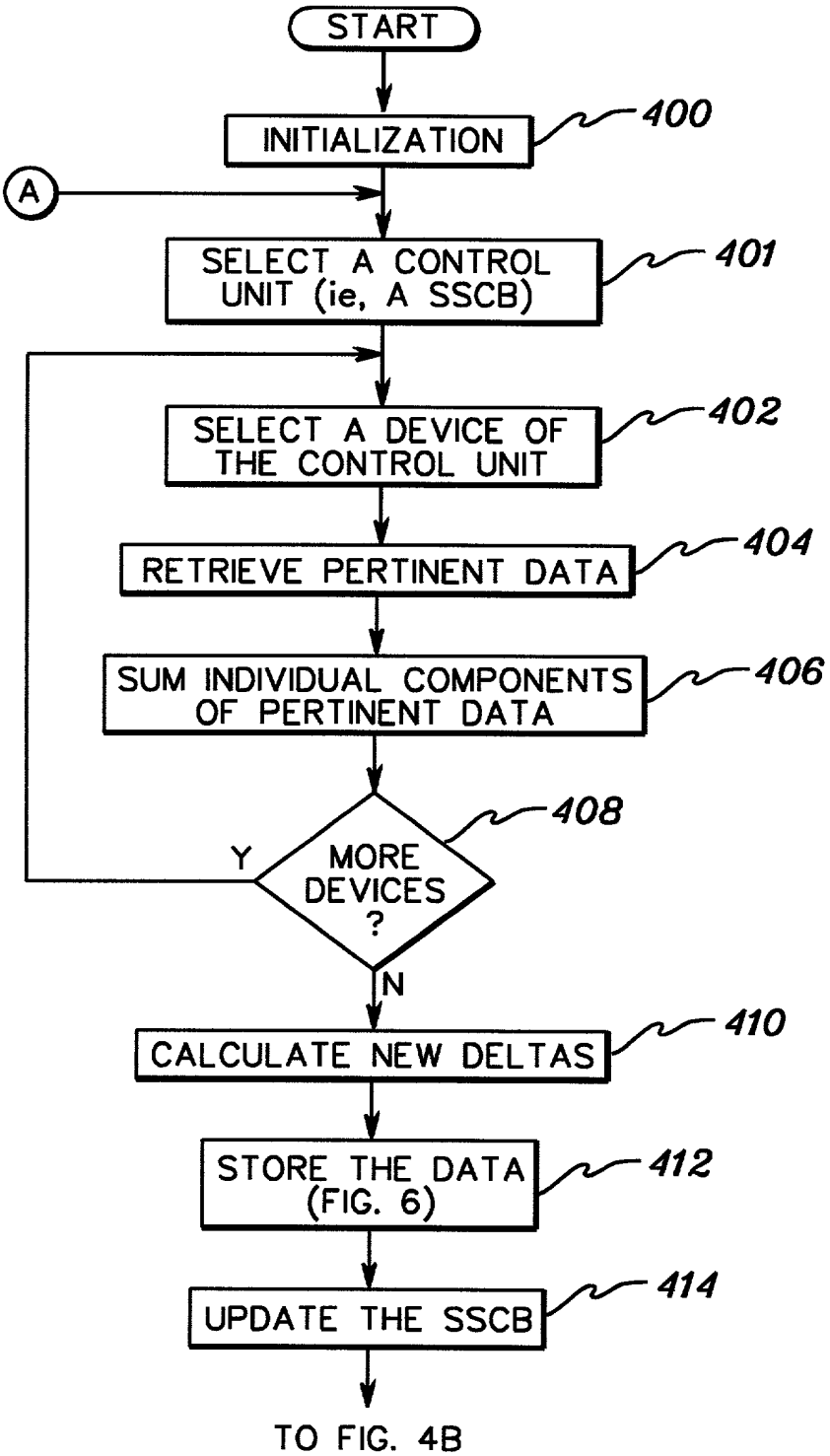
START
INITIALIZATION
400
A
SELECT A CONTROL UNIT (ie, A SSCB)
401
SELECT A DEVICE OF THE CONTROL UNIT
402
RETRIEVE PERTINENT DATA
404
SUM INDIVIDUAL COMPONENTS OF PERTINENT DATA
406
MORE DEVICES ?
408
Y
N
CALCULATE NEW DELTAS
410
STORE THE DATA (FIG. 6)
412
UPDATE THE SSCB
414
TO FIG. 4B

fig. 4A

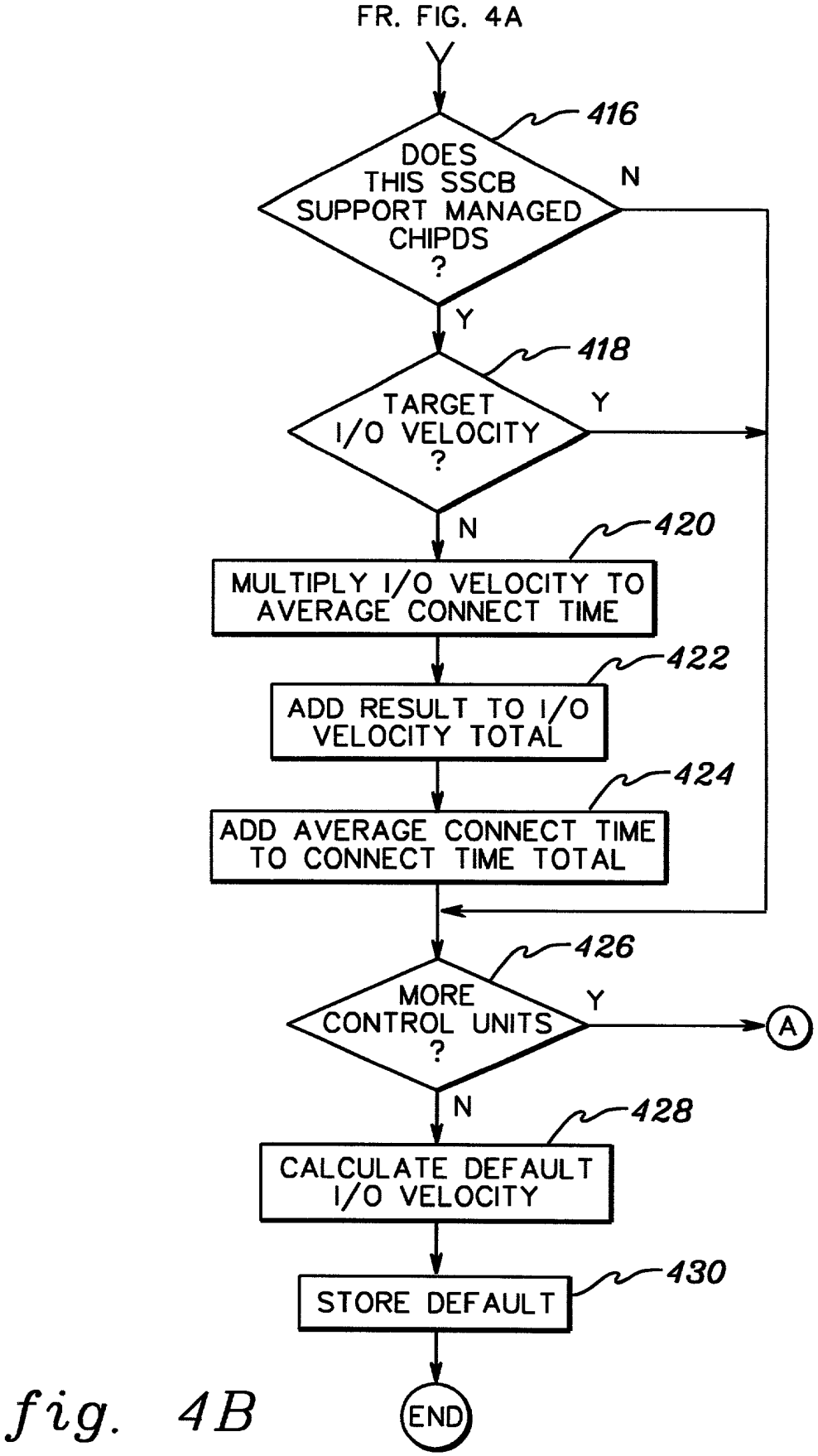

fig. 4B

9607474.092900

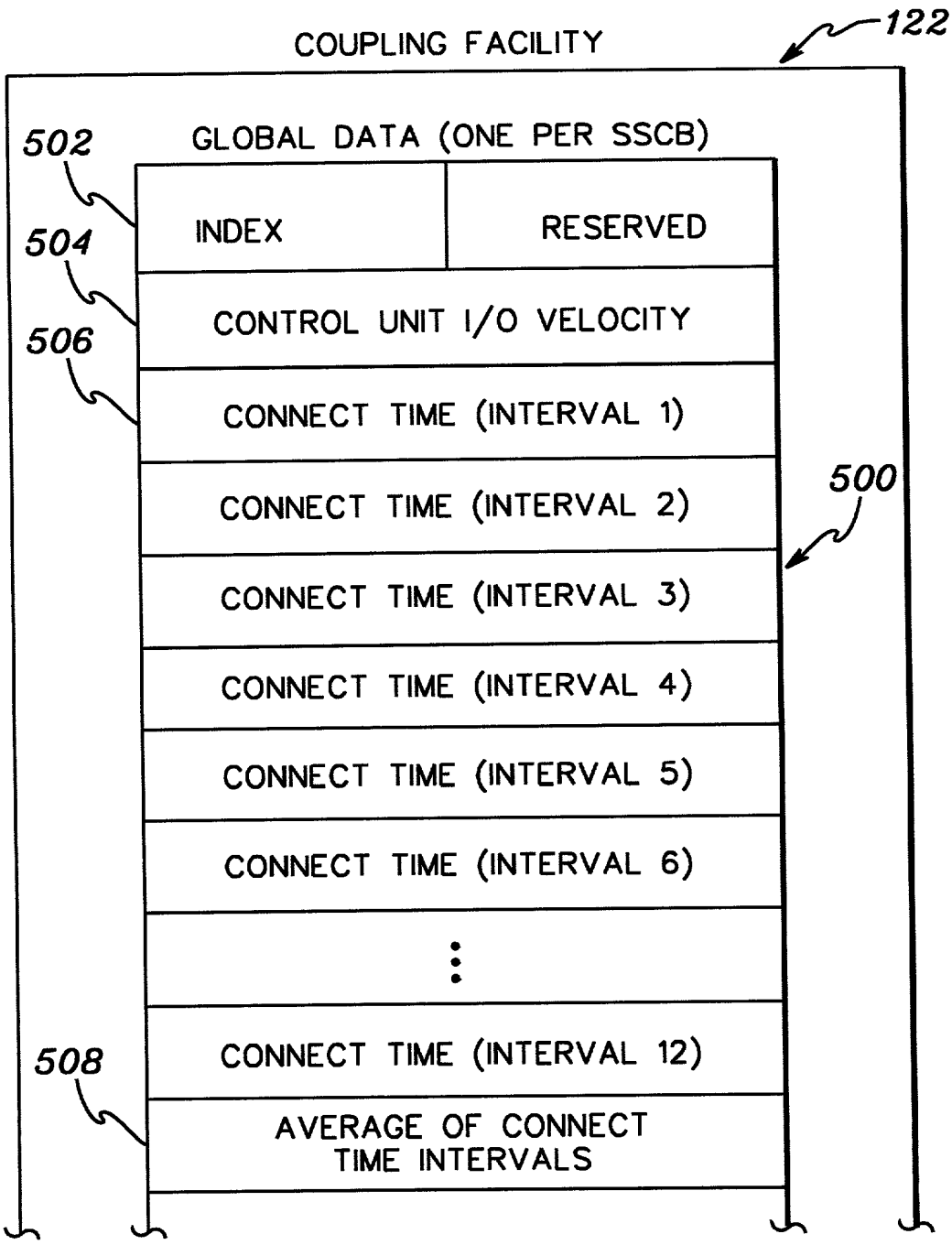
COUPLING FACILITY
122
GLOBAL DATA (ONE PER SSCB)
502
INDEX
RESERVED
504
CONTROL UNIT I/O VELOCITY
506
CONNECT TIME (INTERVAL 1)
CONNECT TIME (INTERVAL 2)
CONNECT TIME (INTERVAL 3)
CONNECT TIME (INTERVAL 4)
CONNECT TIME (INTERVAL 5)
CONNECT TIME (INTERVAL 6)
CONNECT TIME (INTERVAL 12)
508
AVERAGE OF CONNECT TIME INTERVALS
500

fig. 5A

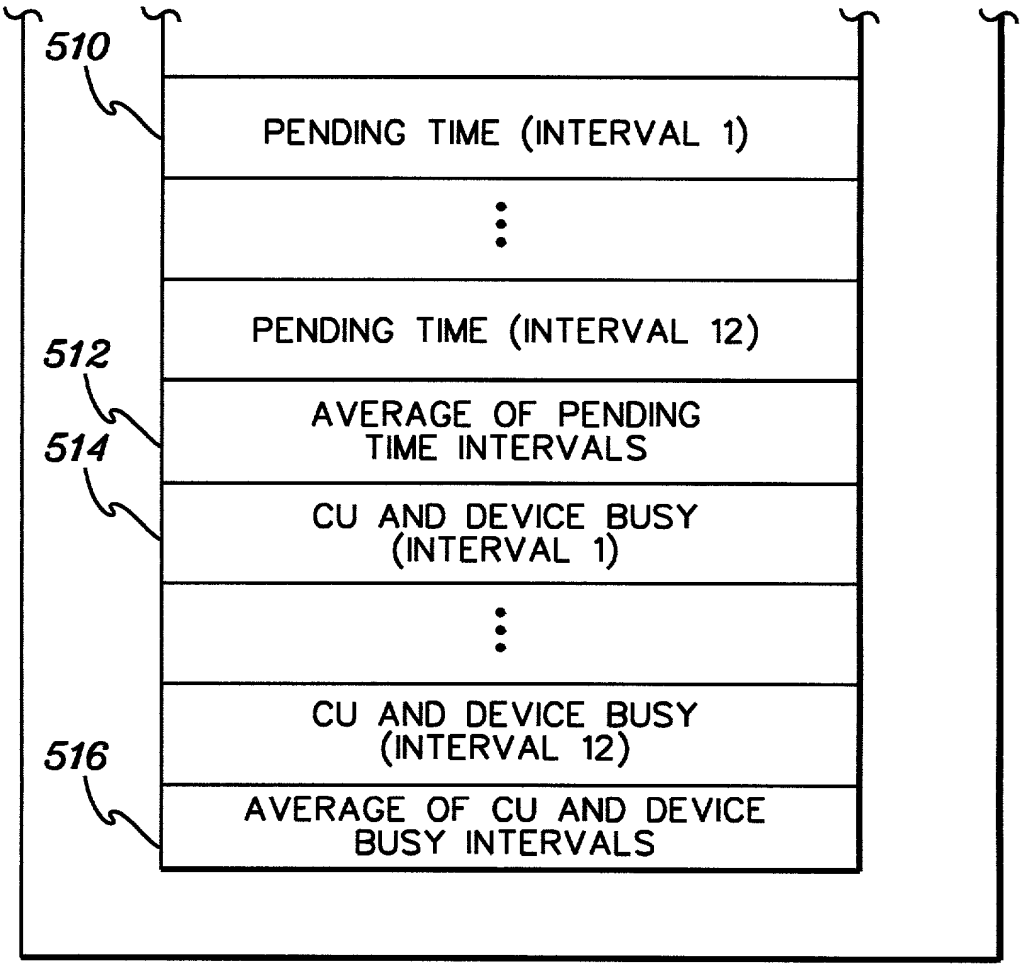
510
PENDING TIME (INTERVAL 1)
PENDING TIME (INTERVAL 12)
512
AVERAGE OF PENDING TIME INTERVALS
514
CU AND DEVICE BUSY (INTERVAL 1)
CU AND DEVICE BUSY (INTERVAL 12)
516
AVERAGE OF CU AND DEVICE BUSY INTERVALS

fig. 5B

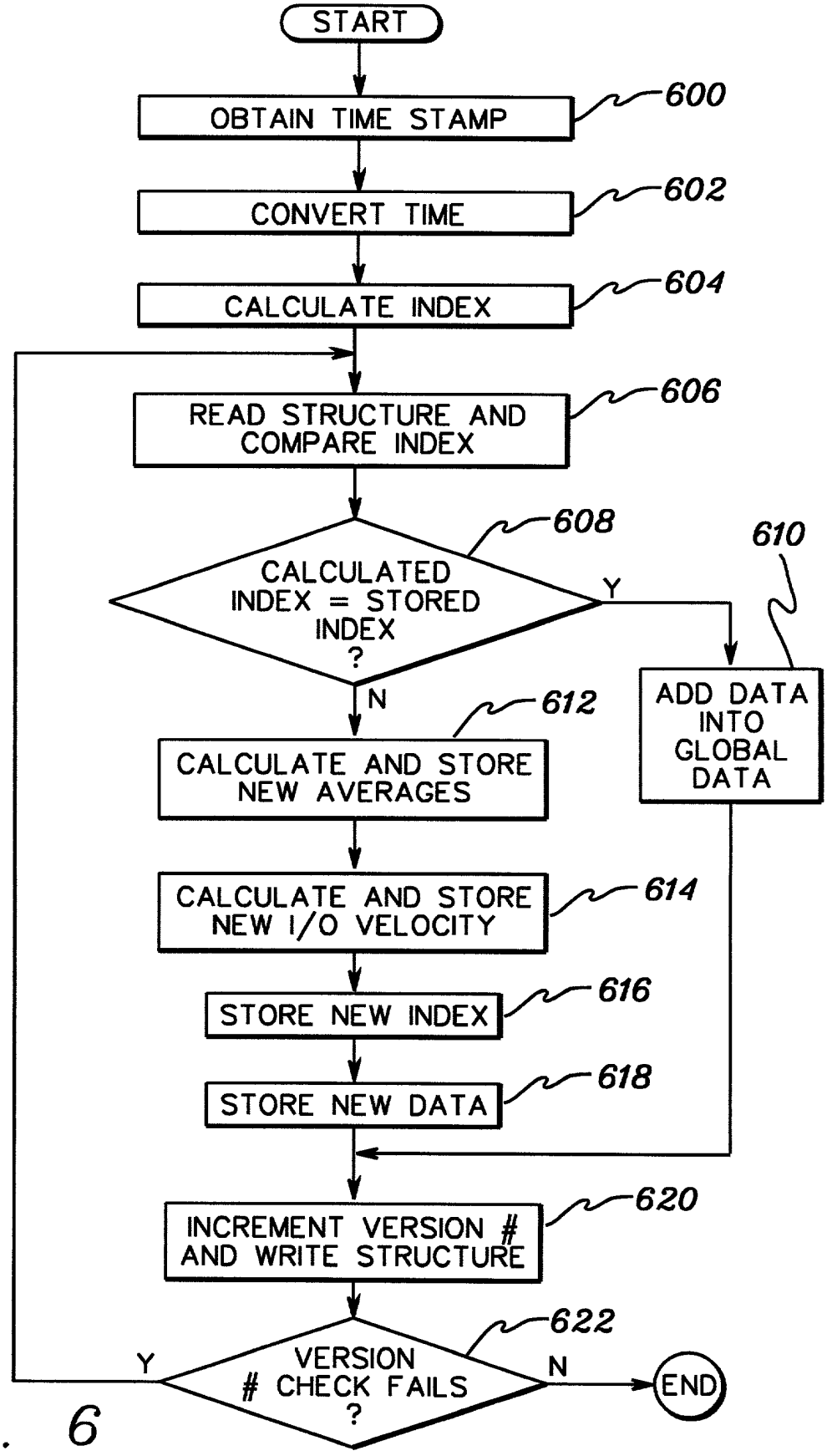
START
OBTAIN TIME STAMP
600
CONVERT TIME
602
CALCULATE INDEX
604
READ STRUCTURE AND COMPARE INDEX
606
CALCULATED INDEX = STORED INDEX ?
608
Y
N
ADD DATA INTO GLOBAL DATA
610
CALCULATE AND STORE NEW AVERAGES
612
CALCULATE AND STORE NEW I/O VELOCITY
614
STORE NEW INDEX
616
STORE NEW DATA
618
INCREMENT VERSION # AND WRITE STRUCTURE
620
VERSION # CHECK FAILS ?
622
Y
N
END

fig. 6

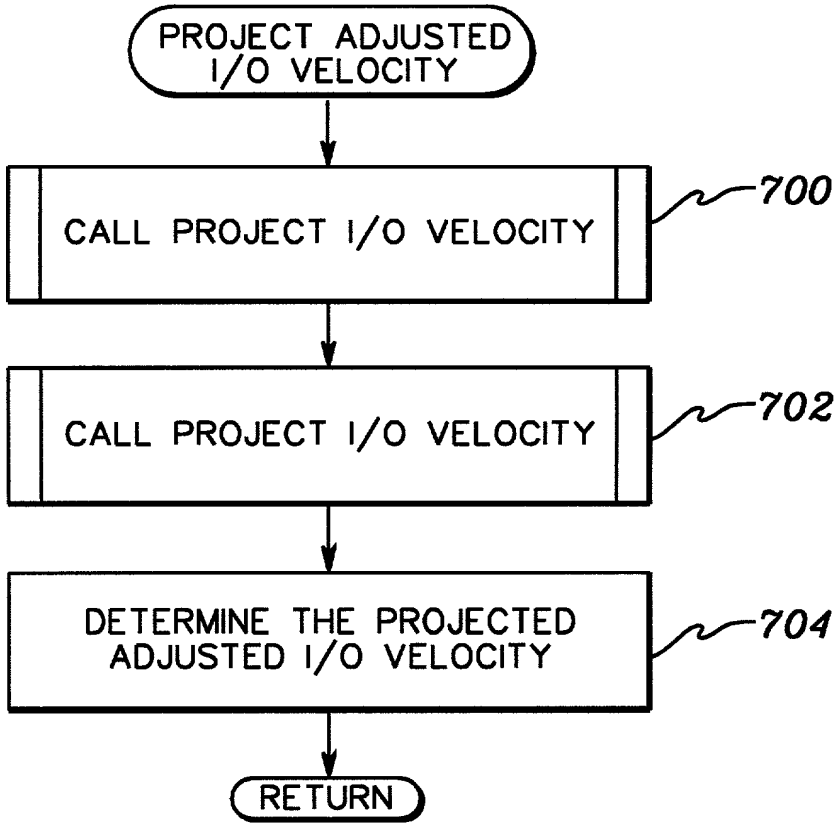
PROJECT ADJUSTED I/O VELOCITY
CALL PROJECT I/O VELOCITY
700
CALL PROJECT I/O VELOCITY
702
DETERMINE THE PROJECTED ADJUSTED I/O VELOCITY
704
RETURN

fig. 7

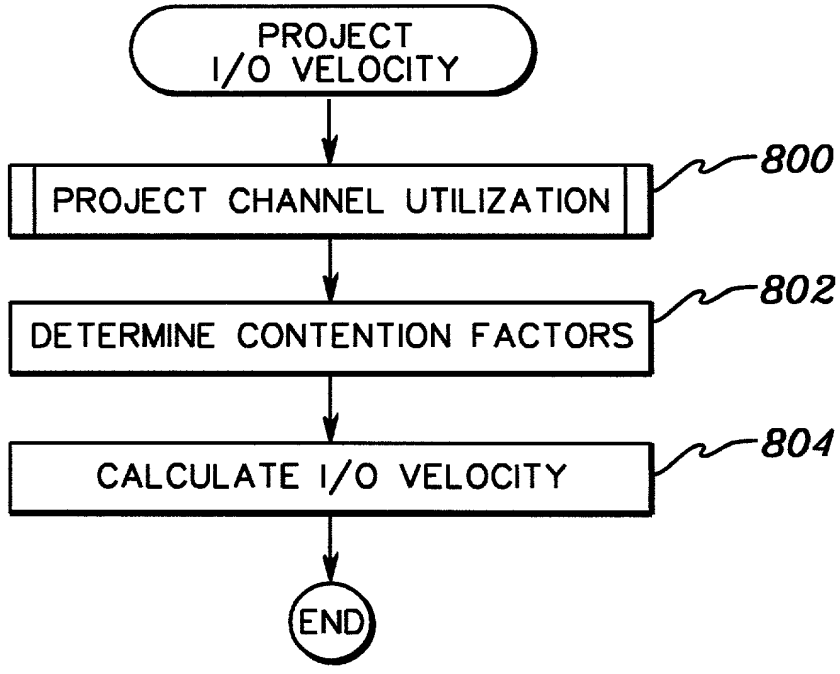
PROJECT I/O VELOCITY
PROJECT CHANNEL UTILIZATION
800
DETERMINE CONTENTION FACTORS
802
CALCULATE I/O VELOCITY
804
END

fig. 8

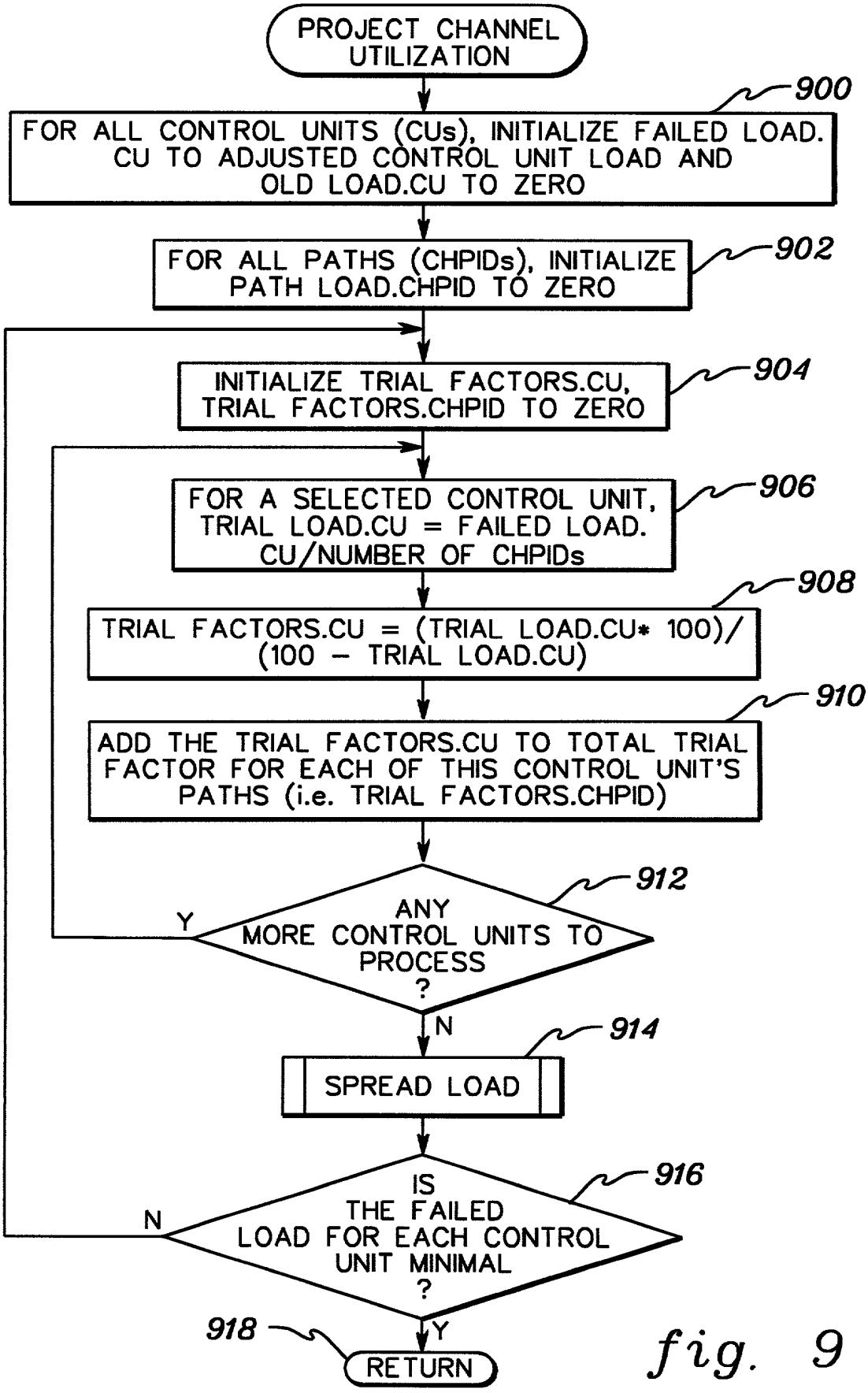
PROJECT CHANNEL UTILIZATION
900
FOR ALL CONTROL UNITS (CUs), INITIALIZE FAILED LOAD. CU TO ADJUSTED CONTROL UNIT LOAD AND OLD LOAD.CU TO ZERO
902
FOR ALL PATHS (CHPIDs), INITIALIZE PATH LOAD.CHPID TO ZERO
904
INITIALIZE TRIAL FACTORS.CU, TRIAL FACTORS.CHPID TO ZERO
906
FOR A SELECTED CONTROL UNIT, TRIAL LOAD.CU = FAILED LOAD. CU/NUMBER OF CHPIDs
908
TRIAL FACTORS.CU = (TRIAL LOAD.CU* 100)/ (100 – TRIAL LOAD.CU)
910
ADD THE TRIAL FACTORS.CU TO TOTAL TRIAL FACTOR FOR EACH OF THIS CONTROL UNIT'S PATHS (i.e. TRIAL FACTORS.CHPID)
912
ANY MORE CONTROL UNITS TO PROCESS ?
Y
N
914
SPREAD LOAD
916
IS THE FAILED LOAD FOR EACH CONTROL UNIT MINIMAL ?
N
Y
918
RETURN

fig. 9

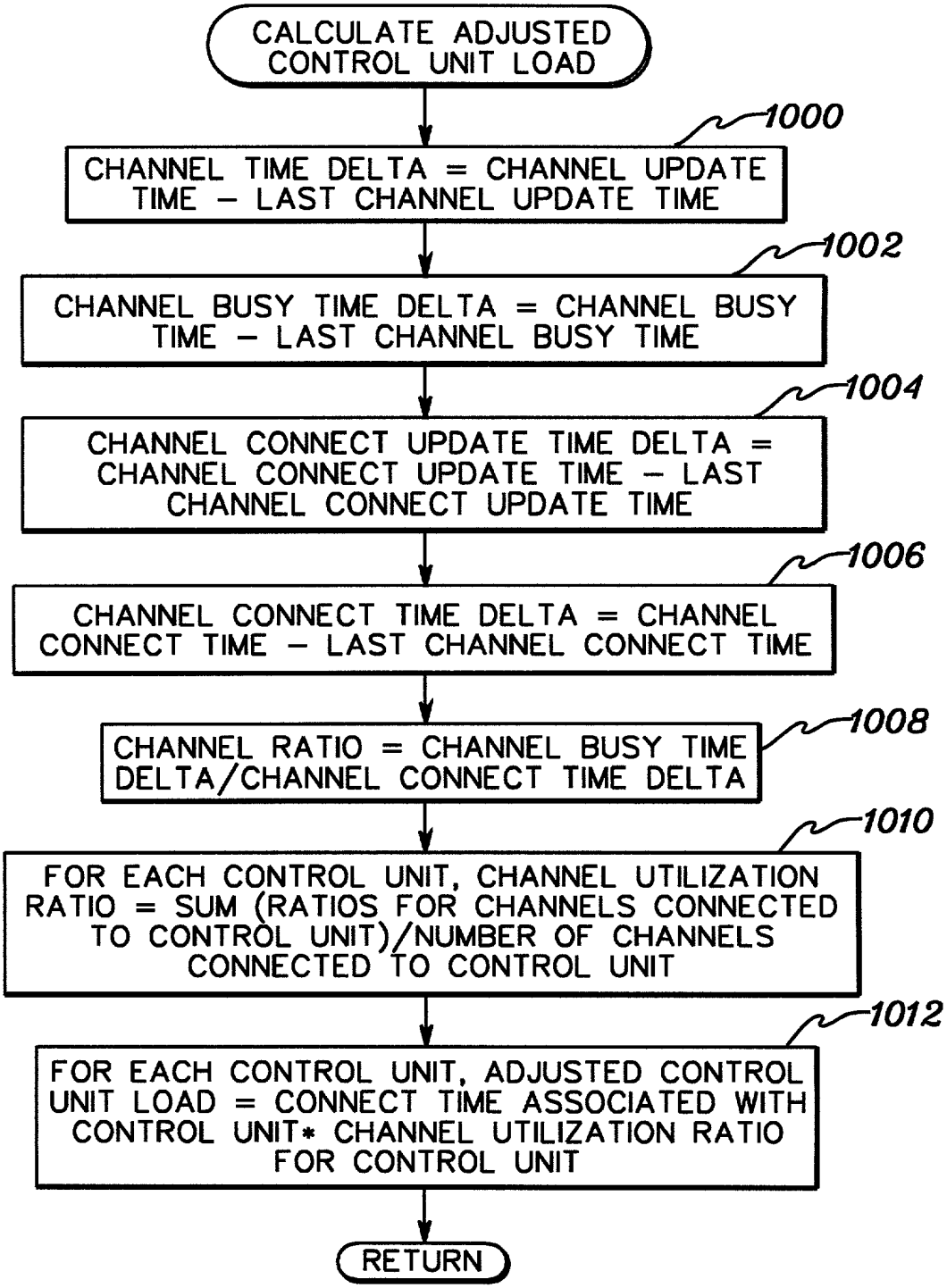
CALCULATE ADJUSTED CONTROL UNIT LOAD
1000
CHANNEL TIME DELTA = CHANNEL UPDATE TIME – LAST CHANNEL UPDATE TIME
1002
CHANNEL BUSY TIME DELTA = CHANNEL BUSY TIME – LAST CHANNEL BUSY TIME
1004
CHANNEL CONNECT UPDATE TIME DELTA = CHANNEL CONNECT UPDATE TIME – LAST CHANNEL CONNECT UPDATE TIME
1006
CHANNEL CONNECT TIME DELTA = CHANNEL CONNECT TIME – LAST CHANNEL CONNECT TIME
1008
CHANNEL RATIO = CHANNEL BUSY TIME DELTA/CHANNEL CONNECT TIME DELTA
1010
FOR EACH CONTROL UNIT, CHANNEL UTILIZATION RATIO = SUM (RATIOS FOR CHANNELS CONNECTED TO CONTROL UNIT)/NUMBER OF CHANNELS CONNECTED TO CONTROL UNIT
1012
FOR EACH CONTROL UNIT, ADJUSTED CONTROL UNIT LOAD = CONNECT TIME ASSOCIATED WITH CONTROL UNIT* CHANNEL UTILIZATION RATIO FOR CONTROL UNIT
RETURN

fig. 10

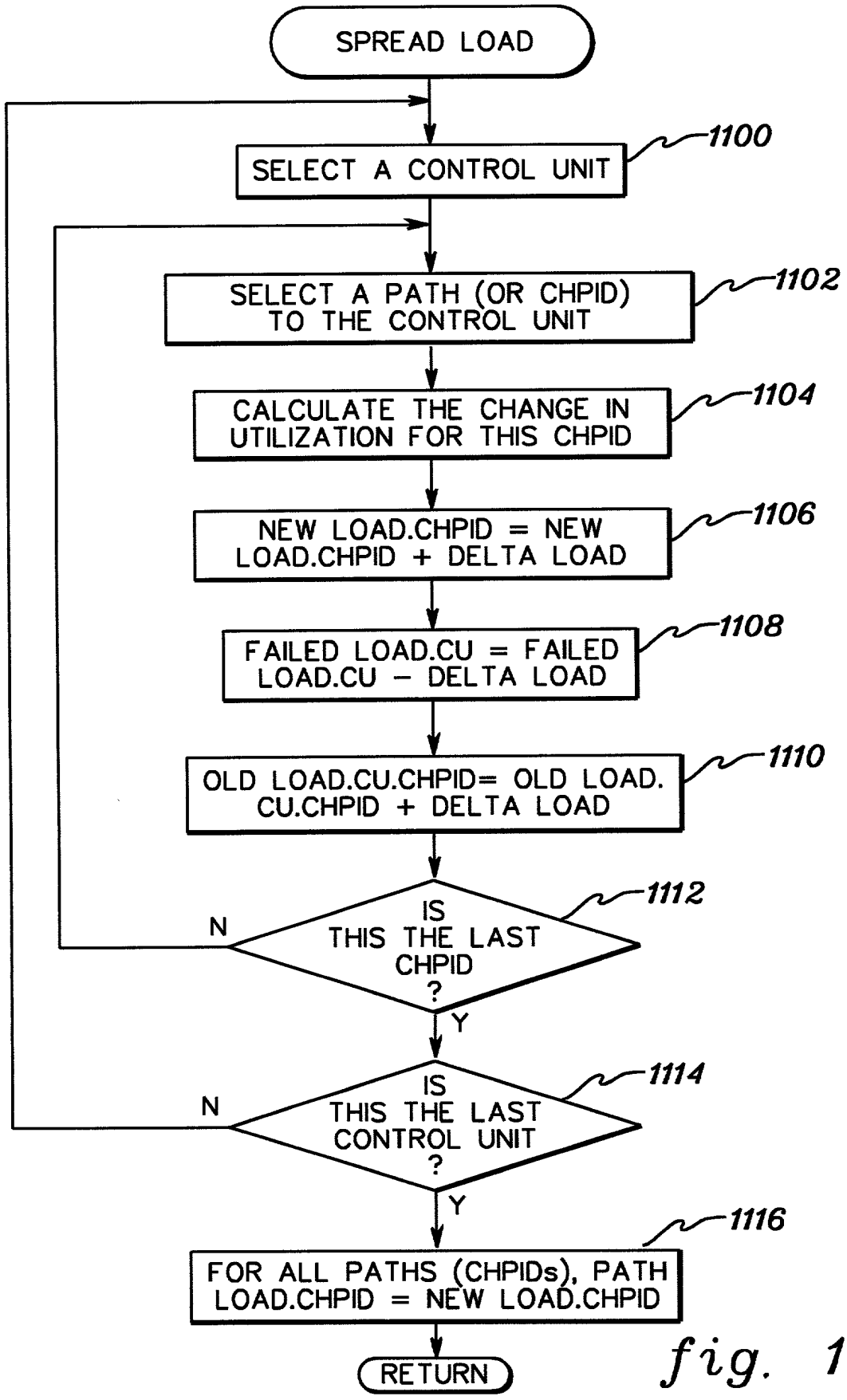
SPREAD LOAD
SELECT A CONTROL UNIT
1100
SELECT A PATH (OR CHPID) TO THE CONTROL UNIT
1102
CALCULATE THE CHANGE IN UTILIZATION FOR THIS CHPID
1104
NEW LOAD.CHPID = NEW LOAD.CHPID + DELTA LOAD
1106
FAILED LOAD.CU = FAILED LOAD.CU – DELTA LOAD
1108
OLD LOAD.CU.CHPID= OLD LOAD. CU.CHPID + DELTA LOAD
1110
IS THIS THE LAST CHPID ?
1112
N
Y
IS THIS THE LAST CONTROL UNIT ?
1114
N
Y
FOR ALL PATHS (CHPIDs), PATH LOAD.CHPID = NEW LOAD.CHPID
1116
RETURN

fig. 11

Docket No.
POU920000130US1

Declaration and Power of Attorney For Patent Application

English Language Declaration

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name,

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

METHOD, SYSTEM AND PROGRAM PRODUCTS FOR DETERMINING WHETHER I/O CONSTRAINTS EXIST FOR CONTROLLERS OF A COMPUTING ENVIRONMENT

the specification of which

(check one)

☒ is attached hereto.

☐ was filed on _____ as United States Application No. or PCT International Application Number _____

and was amended on _____
(if applicable)

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose to the United States Patent and Trademark Office all information known to me to be material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, Section 119(a)-(d) or Section 365(b) of any foreign application(s) for patent or inventor's certificate, or Section 365(a) of any PCT International application which designated at least one country other than the United States, listed below and have also identified below, by checking the box, any foreign application for patent or inventor's certificate or PCT International application having a filing date before that of the application on which priority is claimed.

Prior Foreign Application(s)			Priority Not Claimed
_____ (Number)	_____ (Country)	_____ (Day/Month/Year Filed)	☐
_____ (Number)	_____ (Country)	_____ (Day/Month/Year Filed)	☐
_____ (Number)	_____ (Country)	_____ (Day/Month/Year Filed)	☐

I hereby claim the benefit under 35 U.S.C. Section 119(e) of any United States provisional application(s) listed below:

(Application Serial No.)	(Filing Date)
(Application Serial No.)	(Filing Date)
(Application Serial No.)	(Filing Date)

I hereby claim the benefit under 35 U. S. C. Section 120 of any United States application(s), or Section 365(c) of any PCT International application designating the United States, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT International application in the manner provided by the first paragraph of 35 U.S.C. Section 112, I acknowledge the duty to disclose to the United States Patent and Trademark Office all information known to me to be material to patentability as defined in Title 37, C. F. R., Section 1.56 which became available between the filing date of the prior application and the national or PCT International filing date of this application:

(Application Serial No.)	(Filing Date)	(Status) (patented, pending, abandoned)
(Application Serial No.)	(Filing Date)	(Status) (patented, pending, abandoned)
(Application Serial No.)	(Filing Date)	(Status) (patented, pending, abandoned)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith. *(list name and registration number)*

Lynn L. Augspurger, Reg. No. 24,227
Lawrence D. Cutter, Reg. No. 28,501
Marc A. Ehrlich, Reg. No. 39, 966
William B. Porter, Reg. No. 33,135
Floyd A. Gonzalez, Reg. No. 26,732
William A. Kinnaman, Jr., Reg. No. 27,650
Lily Neff, Reg. No. 38,254
Andrew J. Wojnicki, Jr., Reg. No. 43,995
Christopher A. Hughes, Reg. No. 26,914
Edward A. Pennington, Reg. No. 32,588
John E. Hoel, Reg. No. 26,279
Joseph C. Redmond, Jr., Reg. No. 18,753
Jeff Rothenberg, Reg. No. 26,429
Kevin P. Radigan, Reg. No. 31,789
Blanche E. Schiller, Reg. No. 35,670

Send Correspondence to: **Blanche E. Schiller, Esq.**
HESLIN & ROTHENBERG, P.C.
5 Columbia Circle
Albany, NY 12203

Direct Telephone Calls to: *(name and telephone number)*
Blanche E. Schiller, Esq. **(518) 452-5600**

Full name of sole or first inventor
WILLIAM J. ROONEY

Sole or first inventor's signature — Date

Residence
33 Newhard Place, Hopewell Junction, NY 12533-5004

Citizenship
United States of America

Post Office Address
33 Newhard Place, Hopewell Junction, NY 12533-5004

Full name of second inventor, if any
PETER B. YOCOM

Second inventor's signature — Date

Residence
17B Wildwood Manor, Wappingers Falls, NY 12590

Citizenship
United States of America

Post Office Address
17B Wildwood Manor, Wappingers Falls, NY 12590

Full name of third inventor, if any **HARRY M. YUDENFRIEND**
Third inventor's signature Date
Residence **1 Nob Hill Road, Poughkeepsie, NY 12603**
Citizenship **United States of America**
Post Office Address **1 Nob Hill Road, Poughkeepsie, NY 12603**

Full name of fourth inventor, if any
Fourth inventor's signature Date
Residence
Citizenship
Post Office Address

Full name of fifth inventor, if any
Fifth inventor's signature Date
Residence
Citizenship
Post Office Address

Full name of sixth inventor, if any
Sixth inventor's signature Date
Residence
Citizenship
Post Office Address